



ESCUELA PROFESIONAL DE INGENIERÍA DE TELECOMUNICACIONES

**Extensión del Protocolo ZigBee para un Nuevo
Esquema de Enrutamiento Mediante Filtros de Bloom**

Tesis presentada por el Bachiller en Ingeniería Electrónica y de Telecomunicaciones:

Hector Jesús Ampuero Herrera

Para optar el título profesional de Ingeniero Electrónico y de Telecomunicaciones

Asesor: Dr. **Fernández del Carpio, Gonzalo Mauricio**

Julio, 2016

“Pensar es el trabajo más difícil que existe. Quizá esa sea la razón por la que haya tan pocas personas que lo practiquen.”

Henry Ford, 1863-1947

Índice general

Abstract	9
-----------------	----------

Resumen	11
----------------	-----------

1. Introducción 13

1.1. Motivación y Contexto	15
1.2. Planteamiento del problema	16
1.3. Objetivos	17
1.3.1. Objetivo general	17
1.3.2. Objetivos específicos	17
1.4. Metodología	18
1.5. Consideraciones Complementarias	21
1.5.1. Recursos y Materiales	21
1.5.2. Cronograma de Trabajo	22

2. Marco Teórico 25

2.1. Las Redes de Sensores Inalámbricos (WSN)	25
2.1.1. Características	25
2.1.2. Sensores Inalámbricos	26
2.1.3. Tipos de tráfico	26
2.1.4. Redes de Sensores	27
2.1.5. Uso de energía de los sensores	28
2.1.6. La capa de red	28
2.2. El protocolo IEEE 802.15.4/ZigBee	31
2.2.1. Estándar IEEE 802.15.4	33
2.2.2. Arquitectura	34
2.2.3. Empaquetamiento y Direccionamiento	36
2.2.4. Tipos de dispositivos	36
2.2.5. Topologías	38
2.2.6. Enrutamiento del árbol de ZigBee (ZTR)	39

2.2.7. Tipos de Algoritmos para redes Cluster	41
2.3. Filtros de Bloom	47
2.3.1. Funcionamiento de los filtros de Bloom	48

Índice general

Índice general

i

3. Estado del Arte 53

3.1. Enrutamiento en Redes de Sensores Inalámbricos (WSN) / Capa de Red	53
3.1.1. Algoritmos de Enrutamiento en ZigBee	53
3.1.2. Proceso de enrutamiento	57
3.2. Conclusiones	65

4. Propuesta de Solución 67

4.1. Propuesta de mecanismo de enrutamiento usando Filtros de Bloom .	67
4.2. Formación de árboles multicast	69
4.3. Envío del paquete multicast	72

5. Simulaciones, resultados y comparaciones 75

5.1. Software	75
5.1.1. Eclipse	75
5.1.2. Java	75
5.2. Simulación implementada	76
5.2.1. Generación de ID's de forma aleatoria	77
5.2.2. Descentralización de las transmisiones multicast	78
5.2.3. Simulación de las transmisiones de paquetes en los grupos multicast	78
5.3. Ejecución de las simulaciones	80
5.4. Simulaciones	83
5.5. Resultados	84
5.5.1. Consideraciones	84

5.6. Comparación con el protocolo ZigBee	94
5.7. Conclusiones	100
6. Conclusiones y Trabajos Futuros	103
6.1. Conclusiones	103
6.2. Trabajos Futuros	104
A. Anexo	105
A.1. Anexo de Código fuente	105
Bibliografía	107
Nomenclatura	111

Índice de figuras

1.1. Ámbito de realización del trabajo de investigación [Fuente: Elaboración Propia].	18
1.2. Panorámica de los fundamentos teóricos utilizados [Fuente: Elaboración Propia].	19
1.3. Panorámica del Estado del Arte [Fuente: Elaboración Propia].	19
1.4. Panorámica de Resultados del Sistema ZigBee + Extensión [Fuente: Elaboración Propia].	20
1.5. Panorámica de Funcionamiento de ZigBee [Fuente: Elaboración Propia].	20
1.6. Sistema ZigBee vs. Sistema ZigBee + Extensión [Fuente: Elaboración Propia].	21
2.1. Un ejemplo de un gráfico de unidad de discos (con radio $R/2$) [AN10].	27
2.2. (a) CPU apagada; (b) CPU encendida; (c) Radio encendida; (d) Envío de un paquete a $0dB$; (e) Envío de un paquete a $5dB$; (f) Envío de un paquete a $10dB$; (g) Envío de un paquete a $15dB$ [BG08].	28
2.3. Árbol de Expación Mínima (MST) [AN10].	31
2.4. Aplicaciones ZigBee [C07].	33
2.5. Capas que conforman la pila de protocolos ZigBee [C07].	35
2.6. Campos de los cuatro tipos de paquetes básicos de ZigBee [C07].	37
2.7. Configuración de una red estrella [Mic13a].	38
2.8. Topología árbol o clúster [Mic13a].	38
2.9. Topología Malla [Mic13a].	39
2.10. Estructura de direcciones de ZigBee y relación Padre - Hijo ($C_m 3, R_m 2$, y $L_m 3$) [TK14].	40
2.11. Asignación de direcciones de red [Fuente: Elaboración Propia].	41
2.12. Proceso de selección de cabecera de grupo [EC01].	42
2.13. Enlace configuración entre CH y nodo miembro [EC01].	43
2.14. Estructura de cluster de simple salto [EC01].	43
2.15. Proceso de configuración de múltiple salto de grupo [EC01].	44
2.16. Estructura de múltiple salto de grupo [EC01].	45

2.17. Mensaje HELLO y la actualización de lista de vecinos [EC01].	45
2.18. Lista de vecinos de un nodo en un grupo de borde [EC01].	46
2.19. Reporte de estado de enlace [EC01].	46
2.20. Actualización de topología [EC01].	47
2.21. Inserción de un elemento en un filtro de Bloom [RPL05].	48
2.22. Inserción de dos elemento en el filtro de Bloom.(m=16, h=3)[AR12]. .	49
Índice de figuras	Índice de figuras
<hr/>	
2.23. Ejemplo de filtro de Bloom[TO08].	51
3.1. Mapa conceptual de las técnicas de enrutamiento en WSN y otros	
tipos de redes [Fuente: Elaboración Propia].	54
3.2. Estructura árbol de ZigBee [RP06].	55
3.3. Enrutamiento-Árbol de Acceso Directo (STR) [TK14].	57
3.4. Asignación del CID [JS].	60
3.5. Formación del zFilter y decisión de reenvío en MPSS [GM12]. . .	63
3.6. Ejemplo de construcción de la pila de filtros de Bloom [GM12]. .	64
3.7. Inserción y borrado de un elemento[Fuente: Elaboración Propia]. . . .	65
4.1. Cobertura individual según la potencia de transmisión [Fuente: Elab-	
oración Propia]	68
4.2. Rango de cobertura del nodo R1 [Fuente: Elaboración Propia]. . . .	68
4.3. Asignación aleatoria de un ID a cada nodo y reconocimiento de nodos	
vecinos [Fuente: Elaboración Propia].	69
4.4. Formación de un grupo multicast [Fuente: Elaboración Propia]. . . .	70
4.5. Tipos de nodos en ZigBee [Fuente: Elaboración Propia].	71
4.6. Filtros resúmenes de nodos participantes [Fuente: Elaboración Propia].	71
4.7. Paquete multicast [Fuente: Elaboración Propia]	72
4.8. Verificación de la cabecera del paquete en el nodo R2 [Fuente: Elab-	
oración Propia].	73
4.9. Verificación de la cabecer del paquete en el nodo R1 [Fuente: Elabo-	

ración Propia].	74
5.1. Eclipse [Ecl13].	76
5.2. Radiación de cada nodo [Fuente: Elaboración Propia].	80
5.3. Resultados de nodos vecinos de la red propuesta [Fuente: Elaboración Propia].	81
5.4. Ubicación aleatoria de nodos, red <i>WArequipa</i> [Fuente: Elaboración Propia].	83
5.5. Radio de cobertura de cada nodo [Fuente: Elaboración Propia].	84
5.6. Enlaces virtuales formados por nodos [Fuente: Elaboración Propia].	84
5.7. Ancho de banda útil y desperdiciado, red de 8-nodos [Fuente: Elaboración Propia].	86
5.8. Bucles detectados, red de 8-nodos [Fuente: Elaboración Propia].	87
5.9. Número de enlaces virtuales, red de 8-nodos [Fuente: Elaboración Propia].	88
5.10. Paquetes perdidos por radiación, red de 8-nodos [Fuente: Elaboración Propia].	88
5.11. Bucles detectados, red de 8-nodos [Fuente: Elaboración Propia].	89
5.12. Ancho de banda desperdiciado, red de 8-nodos [Fuente: Elaboración Propia].	89
2	
5.13. Ancho de banda útil usado, red de 8-nodos [Fuente: Elaboración Propia].	90
Índice de figuras	
<hr/>	
5.14. Ancho de banda útil y desperdiciado, red de 14-nodos[Fuente: Elaboración Propia].	91
5.15. Bucles detectados, red de 14-nodos [Fuente: Elaboración Propia].	92
5.16. Número de enlaces, red de 14-nodos[Fuente: Elaboración Propia].	93
5.17. Paquetes perdidos por radiación, red de 14-nodos [Fuente: Elaboración Propia].	93

5.18. Bucles detectados, red de 14-nodos [Fuente: Elaboración Propia]. . . .	93
5.19. Ancho de banda desperdiciado, red de 14-nodos [Fuente: Elaboración Propia].	94
5.20. Ancho de banda útil usado, red de 14-nodos [Fuente: Elaboración Propia].	94
5.21. Asignación de direcciones en redes ZigBee [Fuente: Elaboración Propia].	96
5.22. Asignación de direcciones usando filtros de Bloom [Fuente: Elaboración Propia].	96
5.23. ZigBee con grupos multicast [Fuente: Elaboración Propia].	97
5.24. Uso de filtros de Bloom en la formación de grupos multicast [Fuente: Elaboración Propia].	98
5.25. Envío de un paquete a un grupo multicast en redes ZigBee [Fuente: Elaboración Propia].	99
5.26. Envío de un paquete a un grupo multicast con filtros de Bloom [Fuente: Elaboración Propia].	99
5.27. Total de número de entradas en toda la red [Fuente: Elaboración Propia].	100

Índice de cuadros

1.1. Cronograma de Proyecto de tesis [Fuente: Elaboración Propia].	22
2.1. Tabla comparativa de ZigBee, Bluetooth y Wi-Fi [Fuente: Elaboración Propia].	32
2.2. Bandas de frecuencia del estándar IEEE 802.15.4 [Fuente: Elaboración Propia].	33
2.3. Ventajas y desventajas de la plataforma ZigBee [Zig13].	35
2.4. Tipos de dispositivo IEEE 802.15.4 según su funcionalidad [C07].	37
5.1. Nodos vecinos de la red propuesta en la Figura 5.2 [Fuente: Elaboración Propia].	81
5.2. Filtros resúmenes de nodos hijos [Fuente: Elaboración Propia].	82
5.3. Comparación entre el protocolo ZigBee y la propuesta ZigBee c/Bloom [Fuente: Elaboración Propia].	95
5.4. Valores simulados en relación a los grupos multicast fijos [Fuente: Elaboración Propia].	100

Algoritmos

5.1. Diagrama de flujo de la simulación del envío de un paquete [Fuente: Elaboración Propia].	79
---	----

Abstract

The IEEE 802.15.4/ZigBee standard, designed for wireless sensor networks (WSN), allows WSNs to create their own network topology. That process begins when intermediate and end nodes send messages requesting connection to its parent node. The parent node repeats the request to the coordinator node; then the coordinator node assigns a short address of 16 bits to the nodes, which in turn assigns group addresses to their descendant nodes. The requesting nodes, every so often send their neighboring tables to the coordinator. The coordinator node builds a tree topology, becoming the root of the tree. Thus, all communication must always be done through the coordinator node. This system involves periodic updates of topology tables and records; the address allocation system is complex and requires a calculation of the child nodes, to deliver them the sets of the addresses required. Sensor networks are made up of nodes with low performance and it is desirable to try to minimize the memory usage, processor, energy, etc.

In this work we propose a routing scheme for wireless ZigBee nodes based on Bloom filters. In this proposal, each node independently generates its own identifier, formed as a Bloom filter. For each unicast/multicast request a tree is established and, according to this, summaries of the child nodes are generated and stored in each node. Thus, each packet has as its header the summary of its own path or tree, and hop by hop it is decided to radiate or not radiate the package, depending on whether the verification is successful (made by a simple verification based on AND operations). Although Bloom filters have false positives, in this mechanism they mean a very low percentage, because at each hop, through summaries of nodes, the filter of the packet header is renewed with the summary of the child nodes, avoiding the occurrence of loops. At each node only the summary filter of child nodes are saved per request unicast/multicast. In order to test the effectiveness of this scheme, a great number of multicast requests have been randomly generated for different network topologies. Thus, it has been possible: to eliminate the complexity of the address assignment, the need for the use of large routing tables is reduced and, finally, the packet does not need to be sent always only from the same coordinator node (it can be sent from any node).

Resumen

El estándar IEEE 802.15.4/ZigBee, creado para redes de sensores inalámbricos (WSN), permite a las WSNs crear su propia topología de red. Ese proceso empieza cuando nodos intermedios y finales envían mensajes de solicitud de conexión a su nodo padre. El nodo padre repite la solicitud a su nodo coordinador; luego el nodo coordinador asigna direcciones cortas de 16 bits a los nodos solicitantes, los cuales a su vez asignan grupos de direcciones a sus nodos descendientes. Los nodos solicitantes, cada cierto tiempo envían sus tablas de direcciones vecinas al coordinador. El nodo coordinador forma una topología en árbol, constituyéndose en la raíz del árbol. De esta manera, toda comunicación debe hacerse siempre a través del nodo coordinador. Este sistema implica actualizaciones periódicas de topología, tablas y registros; el sistema de asignación de direcciones es complejo y requiere de un cálculo sobre los nodos hijos, para poder repartirles los conjuntos de direcciones que requieren. Las redes de sensores están conformadas por nodos de bajas prestaciones y es deseable tratar de reducir al máximo el uso de memoria, procesador, energía, etc.

En este trabajo proponemos un esquema de enrutamiento para nodos inalámbricos ZigBee basado en filtros de Bloom. En esta propuesta, cada nodo genera independientemente su propio identificador, formado como un filtro de Bloom. Por cada petición unicast/multicast se establece un árbol y de acuerdo a éste se generan resúmenes de los nodos hijos, que serán almacenados en cada nodo. Así cada paquete tiene como cabecera también el resumen de su propia ruta o árbol, y salto a salto se decide irradiar o no el paquete al llegar a un nodo, dependiendo de si la verificación ha tenido éxito (una verificación sencilla basada en operaciones AND). A pesar que los filtros de Bloom presentan falsos positivos, en este mecanismo éstos significan un muy bajo porcentaje debido a que salto a salto, gracias a los resúmenes de nodos, el filtro de la cabecera del paquete se renueva con el resumen de los nodos hijos, evitando así además la aparición de bucles. En cada nodo únicamente se guarda el filtro resumen de los hijos por cada petición unicast/multicast. Para comprobar la eficacia de este esquema, se han generado aleatoriamente peticiones multicast en distintas topologías de red. De esta manera, se ha conseguido: eliminar la complejidad de la asignación de direcciones, se reduce la necesidad del uso de grandes tablas de enrutamiento y, finalmente, los paquetes no requieren ser enviados siempre desde el mismo nodo coordinador (pueden ser enviados desde cualquier nodo).

1. Introducción

El estándar IEEE 802.15.4/ZigBee surgió como una tecnología para las redes de sensores inalámbricos. Si bien se trata de un sistema inalámbrico, éste no es compatible con otras tecnologías existentes, como Wi-Fi y Bluetooth. ZigBee usa las bandas de frecuencia 868 MHz, 902-928 MHz, y 2.4 GHz. El objetivo de esta tecnología es habilitar redes inalámbricas con capacidades de control y monitoreo que sean confiables, de bajo consumo energético y de bajo costo, por vía radio y de modo bidireccional.

ZigBee es una tecnología muy accesible económicamente, basada en estándares para el control de vigilancia remota y aplicaciones de redes de sensores y control. Hay una serie de aplicaciones de esta tecnología, como la automatización de edificios, sistemas de seguridad, redes de control industrial, entre otros. Debido a su funcionalidad y a la forma en que está diseñada, es posible desarrollar nuevas aplicaciones. A pesar de estos beneficios, ZigBee presenta varias limitaciones, como su baja tasa de transferencia de datos, reducida capacidad de procesamiento, memoria limitada y limitaciones de la energía.

El protocolo ZigBee permite que los dispositivos sean capaces de organizar su propia red, a través de la detección de nodos participantes. La formación de la red comienza cuando un dispositivo se activa y envía solicitudes para unirse a una red. Si recibe una respuesta a su solicitud, éste es unido a la red y se le asigna una dirección pequeña. En caso contrario el nodo se proclama como un nodo coordinador. Luego este nodo envía solicitudes de saludo para detectar a otros dispositivos, y hacerlos parte de su red.

Cuando un nodo distinto al coordinador recibe un paquete de saludo, éste genera una solicitud de unión a la red, mientras éste sigue difundiendo el paquete de saludo. Luego el coordinador registra en tablas de enrutamiento las peticiones de los nodos solicitantes para unirse a la red, y a la vez éste les asigna un direccionamiento corto, para que puedan interactuar en la red. Los nodos pertenecientes a la red, envían periódicamente sus tablas de enrutamiento al nodo coordinador, en las cuales tienen registrados sus nodos vecinos y/o hijos; el nodo coordinador registra estas tablas de enrutamiento, y posteriormente, luego de un análisis, determina un árbol con la ruta más corta para alcanzar a los nodos involucrados en su red.

El nodo coordinador tiene la facultad de llegar a asignar nuevas rutas a los nodos pertenecientes de la red, los cuales en algunos casos determinan nuevos nodos padres e hijos, según la jerarquía árbol de la red. Las redes actuales de ZigBee, generan cantidades inmensas de tablas de enrutamiento, que muchas veces son agregadas de acuerdo a la aplicación a la cual está dirigida, los recursos que caracterizan este protocolo muchas veces son llevados a su límite, como la escasa memoria, lo cual genera factores no favorables para su funcionamiento.

Por otro lado, ZigBee considera dos tipos de conexiones, una es punto a punto y la otra punto a multi-punto. Estas conexiones son realizadas para que los nodos y/o dispositivos puedan interactuar entre sí. Cuando un nodo coordinador envía un mensaje de actualización, éste lo hace muchas veces hacia toda la red, y luego recibe individualmente de cada dispositivo una respuesta. Sería favorable enviar paquetes o solicitudes a dispositivos seleccionados, esto ayudaría a que la red funcione mas adecuadamente, optimizando la energía de los dispositivos y reduciendo el uso del ancho de banda.

Al analizar el tipo de enrutamiento que maneja ZigBee, se ha encontrado algunas debilidades, como la asignación de direcciones que es realizada por un solo nodo previamente seleccionado. Luego de definir un nodo coordinador, la red árbol será formada de acuerdo a sus requerimientos y accesos. Para la realización de algún tipo de comunicación, ésta siempre debe estar unida a un nodo superior, hay nulidad de comunicación directa de nodos, por lo cual es necesario que toda comunicación esté relacionada con un nodo coordinador. Por último, todo este procedimiento que realiza ZigBee genera complejas tablas de enrutamiento, y un mayor uso del ancho de banda.

Hay autores que han generado interés en mejorar el estándar 802.15.4, por lo cual han desarrollado posibles soluciones a este protocolo. En [OG10], para mejorar la comunicación de dispositivos se vio conveniente la agregación de una nueva tabla de enrutamiento, la cual sería agregada en el nodo coordinador y nodos intermedios, además solo contendría información para el establecimiento de una comunicación a un grupo específico de dispositivos. En [Sar12], se nos invita a conocer más sobre la tecnología emergente de redes ZigBee, que está teniendo cada vez un mayor valor en la práctica de algunas redes actuales, que es una de las tecnologías claves de redes Ad-Hoc. En [Sar12], también nos mencionan posibles mejoras que podríamos hacer al estándar 802.15.4, aspectos podríamos mejorar y resultados podríamos alcanzar.

Los autores en [OG10], mencionan que la comunicación en grupo de redes de sensores inalámbricos (WSN), requiere un mecanismo eficiente de enrutamiento de multidifusión, por falta de recursos y por las limitaciones inherentes de computación de los nodos sensores. ZigBee es un protocolo estándar que representa una tecnología muy importante para redes inalámbricas de sensores, no considera el enrutamiento de multidifusión (multicast) en su especificación. En la investigación desarrollada mencionan que un grupo se define como un conjunto de nodos que comparten la misma información sensorial. Las principales contribuciones de este trabajo son dos: Primero, proponen Z-Cast, un mecanismo de enrutamiento multicast eficiente para grupos que comparten la misma información sensorial en un clúster-árbol WSN. En segundo lugar nos muestran la forma de integrar el mecanismo Z-Cast dentro de IEEE 802.15.4/ZigBee.

1.1 Motivación y Contexto

Los autores en [BK07], proponen el modelo multicast basado en dos procedimientos principales: descubrimiento de rutas (RD), y la unión de receptores (RJ). Durante la fase RD un nodo emisor intenta localizar los receptores interesados en una información en especial, y forma una ruta entre ellos. Esta ruta se ve reforzada durante la fase de RJ en el que cada receptor declara su adhesión a la ruta a la fuente de datos. Después se establece la ruta de paquetes de datos multicast, solo para nodos unidos. Por lo tanto la difusión (broadcast) es reducida y el tráfico de red se ve disminuido. También mencionan un estudio sobre los algoritmos modificados, propuestos sobre la base de los algoritmos de encaminamiento existentes, para mejorar el rendimiento de la red.

En el presente documento proponemos un nuevo esquema de enrutamiento que mejorará el enrutamiento de ZigBee, el cual está desarrollado a través del uso de filtros de Bloom, los cuales están integrados de manera independiente en cada nodo o dispositivo. Los nuevos ID's son generados de manera autónoma por cada dispositivo, los cuales son almacenados en filtros de Bloom; ésto permite también que al momento de intercambiar tablas de enrutamiento no ocupen demasiado espacio, además que no tienen la complejidad de ZigBee. También en nuestro nuevo esquema de enrutamiento, proponemos la generación aleatoria de grupos multicast, a través de filtros de Bloom. Este proceso, para una comunicación de este tipo, en ZigBee es muy complejo, mientras que en nuestra propuesta la generación de estos grupos multicast es reducida en un simple filtro de Bloom, haciendo combinatorias solo de nodos seleccionados.

Cuando se generan grupos multicast, la nueva red estará formada solo por nodos seleccionados. Si bien en ZigBee toda comunicación y actualización gira alrededor de un nodo, en nuestra propuesta todos los nodos pueden participar independientemente sobre una red. En este tipo de red los nodos conversan directamente sin necesidad de enviar alguna solicitud al nodo coordinador y/o padre para concretar dicha comunicación.

Los filtros de Bloom han ayudado a muchos tipos de redes, sean cableadas como inalámbricas, mejorar su estructura organizacional y funcional, ya que evita procedimientos complejos, y da mejores resultados. Estas nuevas tablas de enrutamiento permitirán que la sobrecarga sea menor, ayudando al transporte y el establecimiento de comunicación, a la vez reduciendo el consumo de energía y el uso innecesario de la poca memoria que caracteriza a los dispositivos de las WSN.

1.1. Motivación y Contexto

Sería deseable que las redes de sensores tengan métodos de enrutamiento más eficientes para el transporte de paquetes. En el estándar ZigBee, encontramos algunas debilidades en su esquema de enrutamiento, tal como lo hemos descrito anteriormente.

ZigBee es una tecnología emergente que porta múltiples plataformas para desarrollar aplicaciones. Algunos comparan esta tecnología con Wi-Fi o Bluetooth, porque es inalámbrica, pero es distinta en esencia. Está siendo aplicada por grandes industrias, debido a su fácil acceso y bajo costo, a la facilidad de manejo y al amplio rango de aplicaciones que se pueden desarrollar. Surge la necesidad de utilizar más eficientemente los recursos de la red.

El protocolo ZigBee tiene una baja velocidad en transferencia de datos, además cada dispositivo cuenta con una fuente de energía, cuya duración está determinada de acuerdo a la aplicación que se vaya a desarrollar. Esto permitirá un mejor uso y manejo de esta tecnología, ayudara a crear variedad de aplicaciones que no impliquen estructuras complejas y logren obtener información exacta a un menor costo.

El esquema de enrutamiento para el protocolo ZigBee puede ser mejorado de muchas formas, utilizando diferentes tecnologías que optimizan y ayudan a mejorar otros aspectos. El nuevo esquema de enrutamiento propuesto usando filtros de Bloom, permitirá que el protocolo actual pueda reducir el uso innecesario de los recursos limitados, y logre desarrollar nuevas aplicaciones destinadas a más áreas que intensifiquen su uso a un menor costo y complejidad.

1.2. Planteamiento del problema

El problema percibido en ZigBee, más específicamente en redes de Sensores Inalámbricos (WSN), es la falta de un enrutamiento eficiente, sobre todo al entablar comunicación con los miembros de un grupo multicast.

Las redes actuales del estándar ZigBee toman como punto de partida la realización del Spanning Tree (árbol de expansión mínimo). Éste determina el árbol de conectividad mínima hacia los dispositivos y/o nodos involucrados. El árbol formado se mantiene activo a partir de las decisiones tomadas por el nodo coordinador. Toda comunicación, recorre la topología árbol que ha sido determinado por el coordinador.

Además, el nodo coordinador es encargado de asignar direcciones cortas de 16 bits, las cuales son entregadas a los nodos que envían solicitudes para unirse a una red pre-establecida por este nodo coordinador. Los dispositivos y/o nodos solicitantes, generan tráfico y un uso innecesario de la red, cuando hacen este tipo de requerimientos.

En el proceso de actualización, o envío de algún mensaje en específico, esta tecnología centra su enrutamiento en el nodo coordinador, es decir que, si se requiere entablar una comunicación entre dos o más nodos, es necesario mandar la petición hacia el nodo coordinador, y el permitirá la interacción. Esto implica un proceso previo y produce retardo. Ya de por sí, el sistema está diseñado para que no sea posible entablar una

comunicación directa entre nodos; en todos los casos es necesario enviar el paquete al nodo coordinador y éste lo reenvía.

1.3 Objetivos

Otro problema es la limitada memoria de los nodos. Si consideramos, por ejemplo, el envío de paquetes multicast, sería necesario que cada nodo mantenga en su tabla de enrutamiento una entrada por cada grupo multicast y por cada ramificación del árbol de cada grupo. Esto sigue creciendo en complejidad en las partes más altas de los árboles multicast. A medida que el nodo coordinador y los nodos intermedios agregan nuevos nodos a la red, mediante actualizaciones, las tablas crecen.

En resumen, hay cuatro puntos importantes a los que se debe prestar atención y que pueden ser críticos:

- **Spanning tree:** Árbol de expansión mínimo, determinar una red árbol, de acuerdo a la ruta más corta, o menor número de saltos.
- **Asignación de las direcciones:** En ZigBee el nodo coordinador está encargado de asignar direccionamientos cortas de 16 bits.
- **Enrutamiento centralizado en el nodo coordinador:** Toda comunicación viaja primero hacia un nodo coordinador, luego es distribuida según las especificaciones.
- **Tamaño de las tablas de enrutamiento:** Cada actualización del estado de enlace, y agregación de nuevos nodos, son registrados en tablas de enrutamiento.

1.3. Objetivos

1.3.1. Objetivo general

Proponer una mejora al protocolo ZigBee para soportar un nuevo esquema de enrutamiento que codifique la ruta del paquete en su propia cabecera, con el fin de reducir la complejidad y el tamaño de las tablas de enrutamiento que se administran en los nodos sensores. Dicho esquema se plantea que sea conseguido mediante filtros de Bloom.

1.3.2. Objetivos específicos

Los objetivos específicos de este trabajo de investigación son:

1. Investigar los fundamentos teóricos sobre las redes de sensores y el estándar ZigBee.

2. Profundizar en el estado del arte de las soluciones desarrolladas hasta ahora para mitigar las dificultades del enrutamiento de esta tecnología.
3. Implementar una plataforma de simulación para comprobar la propuesta de solución.
4. Analizar los resultados y proponer mejoras futuras a la luz de las comparaciones realizadas.



Figura 1.1.: Ámbito de realización del trabajo de investigación [Fuente: Elaboración Propia].

1.4. Metodología

En este trabajo se propone lograr una extensión del protocolo ZigBee para un nuevo esquema de enrutamiento, mediante el uso de filtros de Bloom. El diseño utilizado permitirá tener un mejor desempeño en el enrutamiento de paquetes en las redes de sensores inalámbricas (WSN).

Para la elaboración de este trabajo, se considera tomar en cuenta el siguiente esquema (ver Figura 1.1). Para el desarrollo de los distintos objetivos específicos propuestos, se realizó consultas de bibliografías disponibles en base de datos, como la IEEE, la Alianza ZigBee, entre otros, cuya información fue descargada para una mejor comprensión. Además también se consultó libros disponibles en la biblioteca de la Universidad.

En los fundamentos teóricos, se realizó una revisión de tres conceptos básicos. Éstos ayudarán a esclarecer los conceptos sobre el funcionamiento de la tecnología ZigBee,

además permitirá conocer los pros y contras de dicha tecnología. En la Figura 1.2 se muestran los temas principales investigados, qué son: las redes de sensores inalámbricas, las especificaciones del estándar 802.15.4, y los conceptos referidos a filtros de Bloom (mecanismo base a través del cual se plantea la mejora en el enrutamiento).

En cuanto al estado del arte, se considera estudiar distintos trabajos propuestos para mejorar el enrutamiento en ZigBee (ver Figura 1.3). El estudio de los trabajos tiene como objetivo entender a profundidad las últimas propuestas, para así tener los elementos necesarios para proponer nuevas soluciones o modificar y mejorar algunas de las que aún están vigentes.

1.4 Metodología

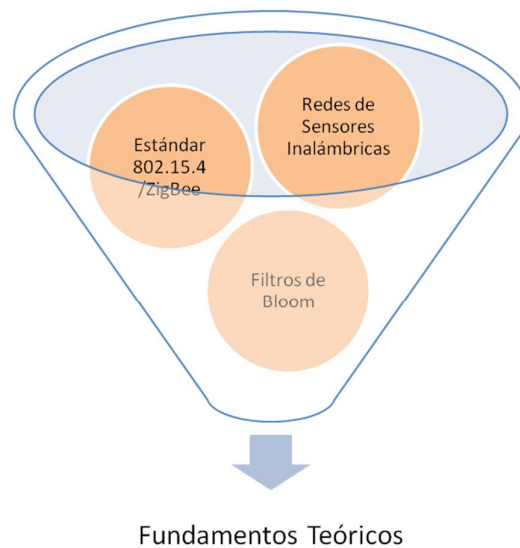


Figura 1.2.: Panorámica de los fundamentos teóricos utilizados [Fuente: Elaboración Propia].

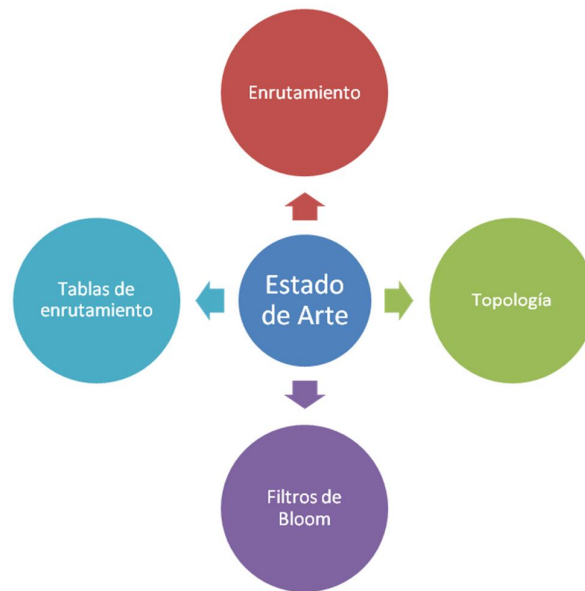


Figura 1.3.: Panorámica del Estado del Arte [Fuente: Elaboración Propia].

Todo lo anterior trae como consecuencia la cristalización de una idea o propuesta de solución, para la extensión del protocolo ZigBee, la cual se detalla en el Cap. IV del presente trabajo. En la Figura 1.4, muestra el procedimiento que se toma en cuenta antes de implementar un nuevo sistema. Para poder probar las prestaciones de la

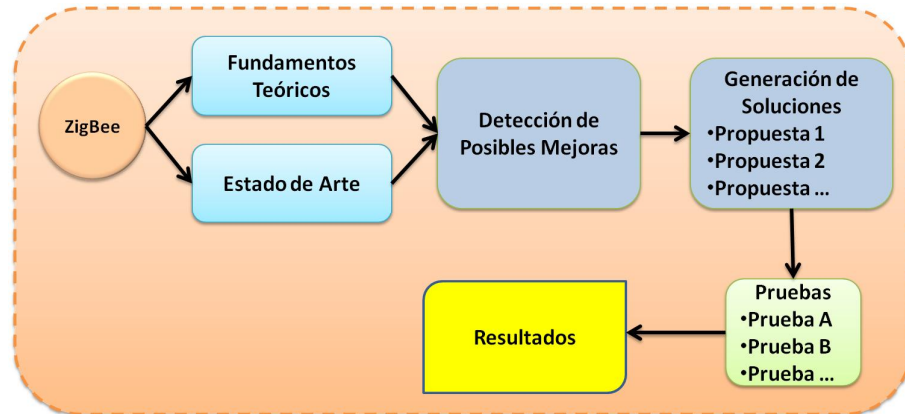


Figura 1.4.: Panorámica de Resultados del Sistema ZigBee + Extensión [Fuente: Elaboración Propia].

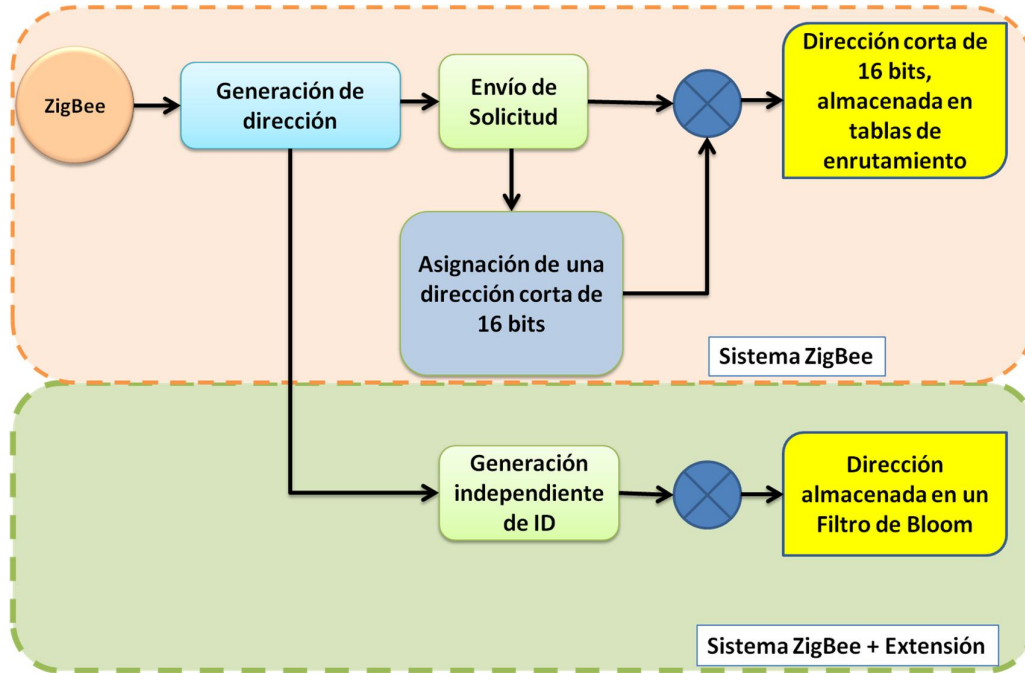


Figura 1.5.: Panorámica de Funcionamiento de ZigBee [Fuente: Elaboración Propia].

nueva propuesta, es necesaria la implementación de una plataforma de simulación, la cual se esquematiza en la Figura 1.5.

Una posible investigación y propuesta a desarrollar, es mostrada en el esquema siguiente, en el cual se realiza la comparación del sistema típico de ZigBee con un nuevo sistema propuesto, el cual trabaja mediante el uso de filtros de Bloom.(Ver

1.5 Consideraciones Complementarias

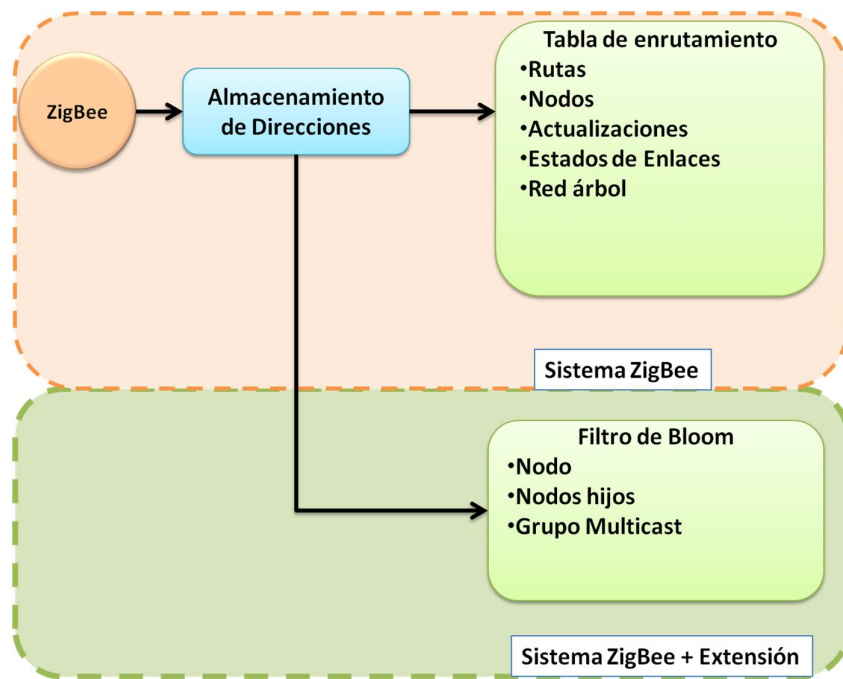


Figura 1.6.: Sistema ZigBee vs. Sistema ZigBee + Extensión [Fuente: Elaboración Propia].

Figura 1.6).

Los resultados esperados se generarán de acuerdo a cuatro parámetros mencionados en el planteamiento del problema (Spanning tree, asignación de direcciones, enrutamiento centralizado en el nodo coordinador y tamaño de tablas de enrutamiento). Los resultados obtenidos y alcanzados permitirán proponer nuevas soluciones al protocolo ZigBee, y qué mejor hacerlo que reutilizando los filtros de Bloom. Ésto se detalla en el Cap. VI.

1.5. Consideraciones Complementarias

1.5.1. Recursos y Materiales

En el trabajo se aplicó conocimientos previos desarrollados a lo largo de la formación académica, se usó un software muy conocido por programadores y desarrolladores de soluciones, el cual nos permitió poder implementar simulaciones de la solución vertida en la documentación adjunta.

Los recursos utilizados son vertidos a continuación:

- Computadora

Actividades	Duración/Semanas	1	2	3	4	5
Asesoría	5	x	x	x		x
Revisión de documentación	4	x	x		x	x
Suposición de problemas	4	x	x		x	x
Programación del algoritmo en Java	3	x	x	x		
Implementación de simulaciones	3		x	x	x	
Pruebas y Resultados	4	x	x	x	x	
Redacción de la tesis	4	x	x	x	x	
Conclusiones y correcciones del informe	2			x	x	
Presentación de tesis	1			x		
Levantamientos de observaciones	3			x	x	x
Presentación final	1					x

Cuadro 1.1.: Cronograma de Proyecto de tesis [Fuente: Elaboración Propia].

- Instalador de Java (JSDK, JRE)
- Instalador de Eclipse (Interfaz de trabajo de Java)

1.5.2. Cronograma de Trabajo

El siguiente cuadro(Cuadro1.1), muestra el cronograma semanal del trabajo de Tesis.

Organización del Presente Documento

La organización del resto del documento es mostrada a continuación.

En el capítulo 2 se brinda información respecto al uso de redes de sensores inalámbricos (WSN), como también la tecnología usada que es ZigBee y tipos de algoritmos para redes Cluster, además se menciona una introducción a filtros de Bloom.

En el capítulo 3 mostramos las investigaciones que hicieron otros autores, y las soluciones que aplicaron para resolver algunos de los problemas relacionados a redes de sensores inalámbricos.

En el capítulo 4 se muestra la propuesta de solución, como también cada paso que se hizo para alcanzar para proponer un nuevo esquema de enrutamiento. Cabe mencionar que al principio hubo dificultades, debido a que cada solución que se proponía era anulada por sus puntos en contra, pero gracias a la investigación previa desarrollada se pudo tener una idea más clara respecto a la idea planteada, y es algo nuevo que muchos no lo han considerado al momento de proponer soluciones en este tipo de redes.

En el capítulo 5 se logra unificar todas las partes, simulando la solución a la que se llegó, la cual presenta menos dificultades y logra obtener grandes beneficios, y todo

1.5 Consideraciones Complementarias

ello puede ser un punto de inicio para poder mejorar cada vez mas las redes ZigBee a través de la solución propuesta y simulada.

2. Marco Teórico

2.1. Las Redes de Sensores Inalámbricos (WSN)

Una red de sensores inalámbrica (WSN), consiste en un conjunto de sensores autónomos distribuidos espacialmente, para monitorear condiciones físicas o ambientales. Cada uno de estos dispositivos o nodos se comunica de manera inalámbrica con otros nodos, de manera bidireccional. Un nodo se puede distinguir por su privilegio de operación y/o funcionamiento, ya sea el nodo principal (o coordinador) que se encuentra constantemente distribuyendo órdenes; y nodos esclavos o finales, quienes ejecutan la petición y regresan respuestas luego de ejecutar las órdenes del nodo coordinador. La mayoría de los datos circulantes son cuantificables; como temperatura, sonido, presión, etc. Por esta razón el crecimiento de estas redes en diversas áreas en los últimos años ha sido muy notable.

La topología de WSN puede variar desde una simple red en estrella a una red de malla inalámbrica. Las técnicas de propagación entre los nodos de la red pueden ser de enrutamiento o de inundación [Dar10][Soh07]. Los puntos importantes a favor de las redes de sensores inalámbricos son el ahorro de energía, el costo de su estructura y la capacidad de los sensores de adaptarse a una infraestructura no preestablecida.

2.1.1. Características

Las características más resaltantes en Redes de Sensores Inalámbricas (WSN) son detalladas a continuación:

- Consumo de energía limitada, por los nodos que utilizan baterías
- Capacidad de afrontar fallos en los nodos
- Movilidad de los nodos
- Heterogeneidad de los nodos
- Escalabilidad a gran escala de despliegue
- En algunos casos soportar duras condiciones ambientales
- Facilidad de uso

2.1.2. Sensores Inalámbricos

Los recientes avances tecnológicos han permitido el desarrollo de sensores de pequeñas dimensiones, de bajo costo, bajo consumo de energía, y dispositivos multifuncionales. Hay diferentes tipos de sensores. Se puede medir la distancia, dirección, velocidad, humedad, velocidad del viento, composición del suelo, temperatura, composición química, luz, movimiento de vibración, actividad sísmica, propiedades acústicas, tensión, torque, carga y presión [AN10].

Tradicionalmente, los sensores están unidos con el medio ambiente, y sus mediciones se envían a una estación base (BS) con comunicación por cable. Los sensores también tienen un pequeño procesador y una memoria pequeña para las señales de codificación y decodificación, así como para el funcionamiento de los protocolos de comunicación simples. Se diferencian en su capacidad de la batería, por ejemplo, algunos de ellos funcionan con baterías pequeñas que duran un mes de funcionamiento continuo, mientras que otros duran un gran periodo de tiempo. En algunas aplicaciones se utiliza una fuente de energía renovable, como un panel solar. Además, algunos sensores están integrados en otros dispositivos y extraen su energía requerida a partir de ellos. En algunos escenarios, los sensores podrían estar provistos de una conexión inalámbrica de un solo salto de acceso a las redes de infraestructura, como Internet.

Para algunas aplicaciones, los sensores pueden ser de un tamaño grande, especialmente si están protegidos por cajas o levantados a una altura que mejora la comunicación y el nivel de protección. Cuando los datos recogidos no son críticos, los sensores pueden funcionar de forma aislada. Por ejemplo, la información sismológica o detección acústica de presencia de aves puede ser simplemente recogida por una memoria de un sensor local y descargada cuando sea necesario [HL10].

2.1.3. Tipos de tráfico

Hay tres tipos de aplicaciones de redes de sensores inalámbricas y cada uno tiene sus modos correspondientes de comunicación de datos: por eventos, periódico, y bajo demanda. En el modo controlado por eventos, los sensores reportan la información detectada a un nodo coordinador una vez que un evento específico ha sido detectado (por ejemplo, un incendio). En el modo periódico (o controlado por tiempo), los nodos sensores reúnen la información del entorno en tiempos predeterminados y periódicamente envían los datos al coordinador. En el modo bajo demanda (controlada por petición), los usuarios deciden cuándo recopilar datos. Envían instrucciones a la

WSN indicando que desea recibir datos y luego espera al tipo requerido de datos que se enviará en el formato solicitado.

2.1 Las Redes de Sensores Inalámbricos (WSN)

2.1.4. Redes de Sensores

La comunicación inalámbrica de múltiples saltos en redes de dispositivos parecidos, se aplican iguales radios de transmisión. Tiene un modelo sencillo que es una simplificación excelente y muy útil de la capa física compleja. En el grafo de disco unitario (UDG), (ver Figura 2.1) dos nodos se comunican si y sólo si la distancia entre ellos es R , donde R es el radio de transmisión que es igual para todos los nodos. Un UDG está determinado por las posiciones de los nodos y un alcance de transmisión común fijo R . Para ilustrar esto, si utilizamos $R/2$ como el radio del disco de cada nodo, dos nodos están conectados si y sólo si sus discos correspondientes cortan o intersecan. Es decir que el radio de cobertura total de un nodo sigue siendo R , lo que permite poder llegar hacia otros nodos, mientras que $R/2$ permite tener intersecciones entre los nodos lo que permiten que estos estén interconectados para su comunicación. Un ejemplo de un UDG se muestra en la Figura 2.1.

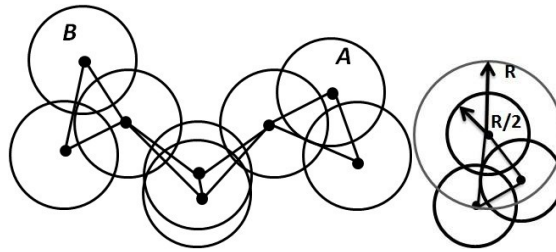


Figura 2.1.: Un ejemplo de un gráfico de unidad de discos (con radio $R / 2$) [AN10].

El número de saltos se puede utilizar como una métrica para encaminar en el UDG si cada nodo aplica la misma potencia de transmisión. Se define como el número de saltos de un nodo a otro. El número de saltos entre dos nodos adyacentes es 1. En la Figura 2.1, el número de saltos entre el nodo A y el nodo B es 4. En redes homogéneas, donde los nodos no ajustan el radio de transmisión, la ruta con menor número de saltos desde un origen a un destino garantiza el coste energético mínimo y la menor latencia de transmisión.

Aunque los protocolos de la capa de red son en su mayoría diseñados con supuestos UDG ideales, los experimentos que se realizan normalmente en los simuladores implementan lo más realista, la capa física y la de control de acceso al medio (MAC). El modelo UDG no es realista, ya que las variaciones en las intensidades de señal recibida

no se consideran. De hecho, se ha señalado que el impacto de las fluctuaciones de intensidad de señal es a veces más significativo que el impacto de la movilidad de los nodos [SI5a]. Por lo tanto, las fluctuaciones deterministas de radio no pueden ser ignoradas en el diseño de protocolos robustos para sensores y redes ad-hoc. Además de la distancia y la intensidad de la señal recibida también dependen de otros factores como el medio ambiente y el medio de transmisión.

2.1.5. Uso de energía de los sensores

El consumo de energía es uno de los principales problemas en redes inalámbricas de sensores. Varios modelos de consumo de energía han sido estudiados, la Figura 2.2 se basa en el estudio de [BG08]. El gráfico muestra el consumo de energía de un nodo pequeño en diferentes estados. El experimento muestra que el cálculo de los costos que recibe el nodo sensor depende de la suposición del estado del nodo. Si se asume que los radios de los nodos están siempre encendidos, el consumo de energía de los costos de recepción es despreciable, ya que el costo para la recepción de paquetes ha sido incluido en el costo de mantener las radios. Más precisamente, el consumo de energía es igual a $2mA$ cuando la radio está apagada, pero es igual a $16mA$ cuando la radio está encendida para recibir. Esto significa que toma alrededor de ocho veces más energía una escucha, en comparación con el estado de hibernación.

Utilizando el ejemplo de la Figura 2.2, se supone que la transmisión de un paquete a $15dB$ consume $60mA$ y toma $5ms$. Recibir el paquete lleva por lo menos $5ms$. Sin embargo, no es posible para el nodo encenderse exactamente en el momento en que se envió el paquete. Es decir, para recibir el paquete, el nodo debe encender su radio durante más de $5ms$. En la Figura 2.2, el consumo de energía de la radio es de $15mA$. Por lo tanto, si el tiempo total que la radio está encendida es más de $20ms$, el consumo de energía en la recepción de un paquete es más que el consumo de energía de transmisión de un paquete [BG08].

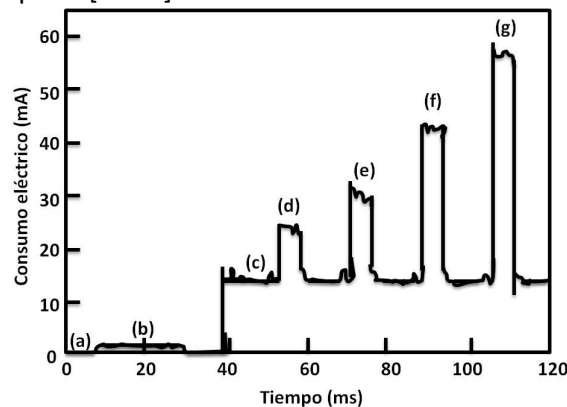


Figura 2.2.: (a) CPU apagada; (b) CPU encendida; (c) Radio encendida; (d) Envío de un paquete a 0dB; (e) Envío de un paquete a 5dB; (f) Envío de un paquete a 10dB; (g) Envío de un paquete a 15dB [BG08].

2.1.6. La capa de red

Los problemas actuales en la capa de red se pueden clasificar en tres categorías: control de topología, enrutamiento y coordinación.

2.1 Las Redes de Sensores Inalámbricos (WSN)

Control de la topología

Una topología de red bien organizada no sólo puede prolongar la vida útil de una red, sino también mejorar las comunicaciones de datos. Los problemas de control de topología pueden subdividirse en problemas de descubrimiento de vecinos y los problemas de organización de la red.

Los problemas de descubrimiento de vecinos se definen como problemas en la detección y descubrimiento de vecinos que se encuentran dentro del rango de cobertura. En la organización de la red, cada nodo elige sus vecinos y construye su topología, ya sea ajustando su potencia de transmisión o ajustando sus estados, como modo reposo y/o modo activo. Hay algunos protocolos que desean lograr la topología de red por control de movimiento en los nodos. Por ejemplo, el protocolo de control de la movilidad local en [DS07] construye una red biconexa desde una red conectada a través del movimiento de los nodos. En la actualidad el control de topología se utiliza para crear redes tolerantes a fallos para protocolos de comunicación fiables.

El control de topología más importante, sobre todo para las redes de sensores críticas en energía, es colocar la mayor cantidad posible de nodos sensores en un modo de espera. Todos los nodos que no son esenciales para la comunicación pueden ser colocados en el modo de reposo durante períodos prolongados, de forma sincrónica o de forma asíncrona. Esto, además de la sincronización de cambios de estado reposoactivo de sensores activos, mejora la eficiencia de energía en la capa MAC. Hay una serie de estudios sobre la eficiencia energética en la capa MAC, que también se basa en el control de topología. S-MAC [YW04] divide los nodos en grupos, basado en los horarios de reposo comunes fijos para reducir la sobrecarga de control y permitir el tráfico de adaptación del despertador. T-MAC [DT03] se extiende de S-MAC mediante el ajuste de la longitud del tiempo de vigilia de los nodos sobre la base de la comunicación de los nodos vecinos. B-MAC [PJ04] emplea un esquema de muestreo preámbulo de adaptación para reducir el ciclo de trabajo y minimizar el tiempo de espera.

Algunos esquemas de control de topología tienen por objeto la selección de determinados nodos de la red para crear una columna vertebral que se puede utilizar de varias maneras. Una columna vertebral se conecta si la red de nodos troncales únicamente permanece conectada[DS07].

La agrupación y el conjunto dominante conectado (CDS) son dos técnicas básicas utilizadas para generar la columna vertebral de sensores inalámbricos y redes ad hoc. El proceso de agrupación divide los nodos de una red en varios grupos. En cada grupo, hay una cabecera de grupo (Cluster head), que es responsable de la coordinación y la comunicación de datos entre los nodos del grupo [DS09]. La selección de la cabecera del grupo se realiza a través de nominación global o elección local, de acuerdo con un protocolo determinado. Las comunicaciones dentro de un grupo podrían ser de un salto o de múltiples saltos.

El nodo coordinador

La red de sensor actuador inalámbrico requiere una coordinación no sólo entre los sensores o actuadores, sino también entre ellos. Para facilitar la coordinación, el primer problema es lograr la selección del actuador adecuado. Los sensores tienen que saber dónde y cómo enviar informes al actuador más cercano. Los actuadores pueden inundar la red con su posición o identidad. Los sensores que reciben dicha información pueden retransmitir a ellos o ignorarlos. Algunos algoritmos de tipo inundación de varios actuadores se han discutido en [IF06]. Una inundación puede descubrir rutas de enrutamiento, los sensores pueden aprender sus caminos hacia el actuador más cercano, en términos de número de saltos u otra medida de distancia. Si la información de posición está disponible y se utiliza en el enrutamiento, la ruta se puede encontrar de forma dinámica.

Los sensores generalmente informan al coordinador o actuador a través de saltos de transmisión. Sin embargo, los actuadores móviles son capaces de moverse para recoger informes periódicamente a través de una ruta prediseñada y por lo tanto minimiza el consumo de energía de los sensores. Por ejemplo, el Cuerpo de Marines de EE.UU. utiliza un vehículo aéreo no tripulado (UAV) para eliminar varios nodos sensores para detectar los vehículos que circulan por una zona desértica [JL03]. Los sensores se organizan para construir una red para monitorear vehículos en movimiento e informar del seguimiento de registros. El avión UAV vuelve más tarde y recupera los datos del seguimiento de cada nodo.

Hay varios problemas asociados con la coordinación de servicios de localización en WSNs [IF06]. En problemas de servicios de localización, los actuadores necesitan de las posiciones de los nodos de sensores para proporcionar y mantener la información de nodos vecinos más cercanos.

El Árbol Mínimo de Expansión

El MST es ejemplo emblemático de las estructuras topológicas de uso común en las redes de comunicación. En un grafo $G = (V, E)$, un MST es un subconjunto conectado del grafo que contiene todos los vértices y minimiza la suma global de longitudes o pesos de sus enlaces. Un algoritmo centralizado bien conocido para construir una estructura de este tipo es el algoritmo de Kruskal, que funciona principalmente como sigue.

Todos los enlaces de E se clasifican en orden creciente de peso, y un nuevo conjunto vacío E' se crea. Para cada arista e en el conjunto ordenado E , se añade e a E' si no crea un bucle en $G' = (V, E')$. El grafo G resultante es un MST de G .

La Figura 2.3 muestra un ejemplo de MST que se ha construido sobre un UDG. Como es usual, en redes inalámbricas, se utiliza la longitud de los enlaces, como su peso. Tenga en cuenta que, si varios enlaces tienen la misma longitud, entonces

pueden existir varios MST equivalentes, pero este no determinismo puede romperse con facilidad sobre la base de parámetros adicionales.

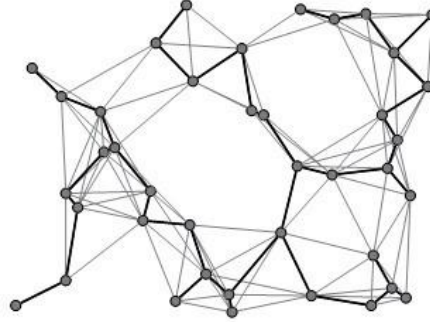


Figura 2.3.: Árbol de Expansión Mínima (MST) [AN10].

Un hecho interesante sobre el MST es que su enlace más largo también se corresponde con el radio de transmisión común óptimo. Esto se deduce directamente del algoritmo de Kruskal, esta ventaja es la última ventaja añadida de E' .

2.2. El protocolo IEEE 802.15.4/ZigBee

Las redes ZigBee se concibieron alrededor de 1998, debido a que las soluciones WiFi y Bluetooth no iban a poder trabajar a bajas tasas de datos, bajo consumo de energía, seguridad y fiabilidad [LC09]. Se puede observar en el Cuadro 2.1 cómo estas características se han concretado en ZigBee, en contraste con las tecnologías Bluetooth y Wi-fi.

El estándar ZigBee se completó el año 2003, y es un estándar para redes inalámbricas de área personal (Wireless Personal Area Network, WPAN). ZigBee ha sido promovido por la ZigBee Alliance, la cual se anunció en el 2004 y es una comunidad internacional de más de 100 compañías como Motorola, Mitsubishi, Philips, Samsung, Honeywell, Siemens, entre otras.

Cuando se concibió este estándar se escogió el término ZigBee en referencia a una colmena de abejas pululando alrededor de su panal y comunicándose entre ellas [C07]. Esta tecnología se basa en estándares para el control de vigilancia remota y aplicaciones de redes de sensores y control. La idea base es mantener una topología de red descentralizada muy similar a la de Internet. Esto permite a los nodos encontrar nuevas rutas a través de la red si una ruta falla, por lo que ese trata de una solución inalámbrica robusta [Zig13].

2.2 El protocolo IEEE 802.15.4/ZigBee

Hay una serie de aplicaciones (Figura 2.4) que se pueden beneficiar del protocolo ZigBee: Redes de sensores, automatización de edificios, sistemas de seguridad, redes de control industrial, medición remota y periféricos de ordenadores.

Estándar	Wi-Fi802.11g	Wi-Fi802.11b	Bluetooth802.15.1
Aplicacionesprincipales	WLAN	WLAN	WPAN(sustituir cableentre dispositivos)
Memoriainecesaria	MB	MB	0 KB
VidaBatería(días)	0.5-5	0.5-5	1-7
TamañoRed	nodec	nodec	7
Velocidad(kbps)	Mb/sec	Mb/sec	10(v1.1)
Cobertura(metros)	100	100	10(v1.1)
Parámetrosmsimportantes	VelocidadyFlexibilidad	VelocidadyFlexibilidad	Costeyperfilesdeaplicación
Frecuenciadeoperación	4 GHz	4 GHz	4 GHz
TopologíadeRed	Arbol,estrella,malla,anillo,bus,línea	Arbol,estrella,malla,anillo,bus,línea	Picoredes

Cuadro2.1.: TablaComparativadeZigBee,BluetoothyWi-FiFuente:ElaboraciónPropia1.

2.2 El protocolo IEEE 802.15.4/ZigBee

ZigBee802.15.4
Control y monitorización
KB-32KB
100-1000+
255/65000
250
1-100
Fiabilidad, bajo consumo y muy bajo coste
Más de 1000000 de dispositivos
Arbol, estrella, malla



Figura 2.4.: Aplicaciones ZigBee [C07].

En comparación con otros protocolos inalámbricos, el protocolo ZigBee requiere una baja complejidad, reduciendo las necesidades de recursos y además ofrece un conjunto estándar de especificaciones. También tiene un número de configuraciones de red y capacidad de seguridad opcional [C07][Zig13].

2.2.1. Estándar IEEE 802.15.4

El protocolo ZigBee utiliza la especificación IEEE 802.15.4 como su capa de acceso al medio (MAC) y física (PHY). Proporciona una metodología estándar para las funciones, incluyendo la formación de redes de mensajería y detección de dispositivos. El estándar IEEE 802.15.4 opera en tres bandas de frecuencia (ver Cuadro 2.2) y cada una tiene un número fijo de canales [LC09].

Frecuencia	Número de Canales	Tasa de transmisión de bits
2.4 GHz	16 Ch (11-26)	250 Kbps
915 MHz	10 Ch (1-10)	40 Kbps
868 MHz	1 Ch (0)	20 Kbps

Cuadro 2.2.: Bandas de frecuencia del estándar IEEE 802.15.4 [Fuente: Elaboración Propia].

La longitud máxima de un paquete MAC IEEE 802.15.4 es de 127 bytes, incluyendo un valor CRC (Cyclic Redundancy Check) de 16 bits que comprueba la integridad de la trama.

2.2 El protocolo IEEE 802.15.4/ZigBee

Además, opcionalmente utiliza un mecanismo de transferencia de datos reconocidos. En resumen, las principales características de este estándar son:

- Ha sido desarrollado como un estándar abierto mundial para hacer frente a las necesidades de bajo costo y bajo consumo de redes inalámbricas.
- Opera en las bandas libres ISM¹ (Industrial, Scientific & Medical) (Ver Cuadro 2.2).
 - Los rangos de cobertura son de 10 a 75 m.
 - Una red ZigBee puede estar formada por hasta 255 nodos.
 - A pesar de coexistir en la misma frecuencia con las redes Wi-fi y Bluetooth, su desempeño no se ve afectado, debido a su baja tasa de transmisión y a características propias del estándar IEEE 802.15.4.
- Soporta diferentes tipos de topologías como estrella, punto a punto, malla y árbol.
- Es un protocolo de comunicación de múltiples saltos, es decir, que se puede establecer comunicación entre dos nodos aún cuando estos se encuentren fuera del rango de transmisión [LC09].

ZigBee/IEEE 802.15.4 maneja tres tipos de tráfico típicos [LC09]:

- Cuando el dato es periódico: La aplicación envía una orden, el sensor se activa, chequea los datos y luego se desactiva.
- Cuando el dato es intermitente: La aplicación u otro estímulo, determina el tiempo de activación del interfaz de red.
- Cuando el dato es repetitivo: El tiempo es prefijado dependiendo de los espacios de tiempo repartidos, los dispositivos operan en duraciones fijas.

En general, la plataforma ZigBee presenta las siguientes ventajas y desventajas (Ver Cuadro 2.3):

2.2.2. Arquitectura

ZigBee es una pila de protocolos, que de manera similar al modelo OSI, está constituido por diferentes capas independientes unas de otras mostradas en la Figura 2.5 [C07]. ZigBee y 802.15.4. son a veces utilizados como sinónimos cuando en realidad definen partes diferentes de la pila de protocolos [Gar09].

- La capa de más bajo nivel es la capa física (PHY), que junto con la capa de acceso al medio (MAC), brindan los servicios de transmisión de datos por el aire, punto a punto. Estas dos capas están descritas en el estándar IEEE 802.15.4–2003

¹ ISM: Industrial, Scientific & Medical

[Soc13][C07][CA008]. Tal como se mencionó previamente, el estándar trabaja sobre las bandas ISM de uso no regulado, dónde se definen hasta 16 canales en el rango de 2.4 GHz, cada una de ellas con un ancho de banda de 5 MHz.

Ventaja	Desventaja
Ideal para conexiones punto a punto y punto a multipunto, por su sistema de múltiples saltos.	Protocolo de enrutamiento muy complejo e ineficiente.
Es muy adecuado para redes de baja tasa de transferencia de datos.	La tasa de transferencia es muy baja.
Bajo ciclo de trabajo, lo cual proporciona larga duración de la batería.	Sólo manipula mensajes pequeños en comparación con otras tecnologías.
Soporte para múltiples topologías de red: Estática, dinámica, estrella y malla.	No es compatible con Bluetooth.
Soporta hasta 65.000 nodos en una red.	Tiene menor cobertura, aunque es extendible a través de los múltiples saltos.

Cuadro 2.3.: Ventajas y desventajas de la plataforma ZigBee [Zig13].

Aplicación/Perfil	Usuario
Soporte de Aplicación	ZigBee Alliance
Red(NWK)/Seguridad (SSP)	
MAC	IEEE 802.15.4
PHY	

Figura 2.5.: Capas que conforman la pila de protocolos ZigBee [C07].

- La capa de red (NWK), tiene como objetivo principal permitir el uso correcto del subnivel MAC y ofrecer una interfaz adecuada para su uso por parte de la capa de aplicación. En esta capa se brindan los métodos necesarios para: iniciar la red, unirse a la red, encaminar paquetes dirigidos a otros nodos en la red, proporcionar los medios para garantizar la entrega del paquete al destinatario final, filtrar paquetes recibidos, cifrarlos y autentificarlos. Cuando esta capa se encuentra cumpliendo la función de unir o separar dispositivos a través del controlador de red, implementa seguridad y encamina tramas a sus respectivos destinos; además, la capa de red del controlador de red es responsable de crear una nueva red y asignar direcciones a los dispositivos de la misma. Es en esta

2.2 El protocolo IEEE 802.15.4/ZigBee

capa donde se implementan las distintas topologías de red que ZigBee soporta (árbol, estrella y red mallada).

- La siguiente capa es la de soporte de aplicación. Es la responsable de mantener el rol que el nodo juega en la red, filtrar paquetes a nivel de aplicación, mantener la relación de grupos y dispositivos con los que la aplicación interactúa y simplificar el envío de datos a los diferentes nodos de la red.
- En el nivel conceptual más alto se encuentra la capa de aplicación, que no es otra cosa que la aplicación misma. Es en esta capa donde se encuentran los ZDO (ZigBee Device Objects) que se encargan de definir el papel del dispositivo en la red, si éste actuará como coordinador, encaminador o dispositivo final.

Cada capa se comunica con sus capas subyacentes a través de una interfaz de datos y otra de control, las capas superiores solicitan servicios a las capas inferiores, y éstas reportan sus resultados a las superiores [C07].

2.2.3. Empaquetamiento y Direccionamiento

En ZigBee, el empaquetamiento se realiza en cuatro tipos diferentes de paquetes básicos mostrados en la Figura 2.6, los cuales son: Datos, ACK, MAC y baliza [C07].

- El paquete de datos tiene una carga de hasta 104 bytes. La trama está numerada para asegurar que todos los paquetes lleguen a su destino. Un campo CRC asegura que el paquete se ha recibido sin errores. Esta estructura aumenta la fiabilidad en condiciones complicadas de transmisión.
- Los paquetes ACK, llamados también paquete de reconocimiento, permiten una realimentación desde el receptor al emisor, de esta manera se confirma que el paquete se ha recibido sin errores.
- El paquete MAC se utiliza para el control remoto y la configuración de dispositivos/nodos. Una red centralizada utiliza este tipo de paquetes para configurar la red a distancia.
- El paquete baliza se encarga de “despertar” los dispositivos que “escuchan” y luego vuelven a “dormirse” si no reciben nada más. Estos paquetes son importantes para mantener todos los dispositivos y los nodos sincronizados, sin tener que gastar una gran cantidad de batería estando todo el tiempo encendidos.

2.2.4. Tipos de dispositivos

IEEE 802.15.4 define dos tipos de dispositivos según su funcionalidad, que se muestran en el Cuadro 2.5:

- Dispositivo de funcionalidad completa (FFD): También conocido como nodo activo. Gracias a la memoria adicional y a su capacidad de computación puede funcionar como Coordinador o Router ZigBee.
- Dispositivo de funcionalidad reducida (RFD): También conocido como nodo pasivo. Tiene capacidad y funcionalidad limitadas con el objetivo de conseguir

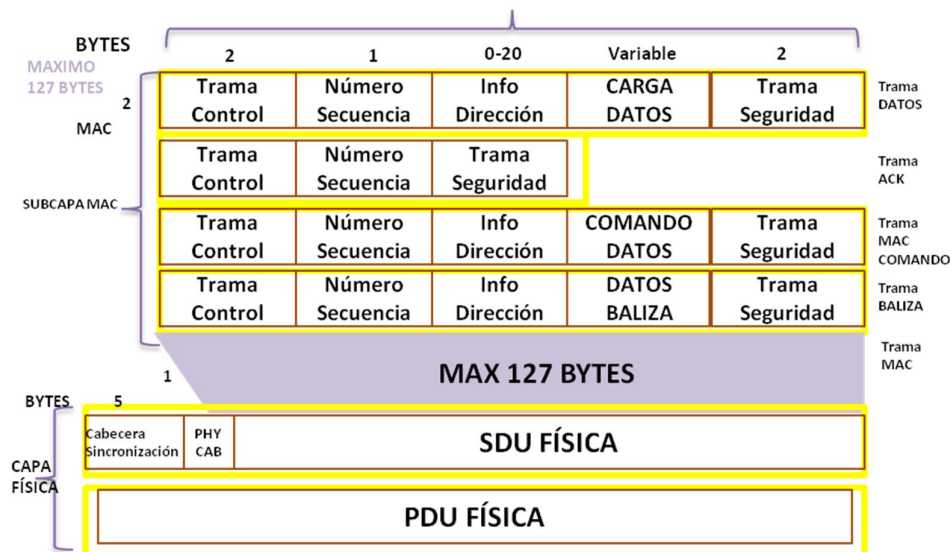


Figura 2.6.: Campos de los cuatro tipos de paquetes básicos de ZigBee [C07].

un bajo coste y una gran simplicidad. Básicamente son los sensores/actuadores de la red [C07][CAO08][LC09].

Tipo de dispositivo	Servicio que ofrece	Fuente de energía	Configuración típica
Dispositivo de funcionalidad completa (FFD)	La mayoría o todos	Red eléctrica	ON, cuando está en reposo
Dispositivo de funcionalidad reducida (RFD)	Limitado	Baterías	OFF, cuando está en reposo

Cuadro 2.4.: Tipos de dispositivo IEEE 802.15.4 según su funcionalidad [C07].

Se definen tres tipos diferentes de dispositivos ZigBee según su papel en la red:

- Coordinador ZigBee (ZigBee Coordinator, ZC): Es el tipo de dispositivo más completo, puede actuar como director de una red en árbol, así como servir de

2.2 El protocolo IEEE 802.15.4/ZigBee

enlace a otras redes. Existe exactamente un coordinador por cada red [CAO08][C07].

- Router ZigBee (ZR): Además de ofrecer un nivel de aplicación para la ejecución de código de usuario, puede actuar como encaminador, interconectando dispositivos separados en la topología de la red.
- Dispositivo final (ZigBee End Device, ZED): Posee la funcionalidad necesaria para comunicarse con su nodo padre (el coordinador o un encaminador), pero no puede transmitir información destinada a otros dispositivos. De esta forma, este tipo de nodo puede estar dormido la mayor parte del tiempo, aumentando la vida media de sus baterías [C07].

2.2.5. Topologías

Como ya lo mencionamos anteriormente, ZigBee permite la configuración de las siguientes distintas topologías:

- Topología en estrella: Consiste en un nodo ZigBee coordinador y uno o más dispositivos finales. En una red en estrella, todos los dispositivos finales se comunican sólo con el coordinador (Ver Figura 2.7) [Mic13a].

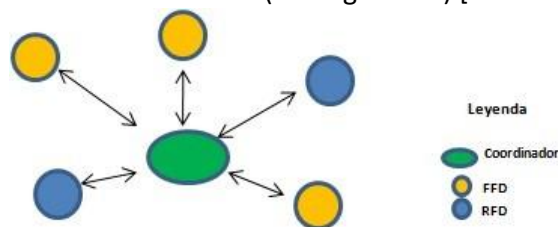


Figura 2.7.: Configuración de una red estrella [Mic13a].

En la sección anterior se menciona que FFD (Dispositivo de funcionalidad completa), puede asumir funciones como nodo Coordinador, o nodo Router (Enrutamiento), mientras que RFD (Dispositivo de funcionalidad reducida) asume la función como nodo final.

- Topología en árbol o cluster: En esta configuración los dispositivos finales pueden unirse al coordinador ZigBee o a un encaminador(router) ZigBee, como lo muestra en la Figura 2.8. Los encaminadores sirven para dos funciones: Una es aumentar el número de nodos que pueden estar en una red, y la otra es para ampliar el rango físico de la red.

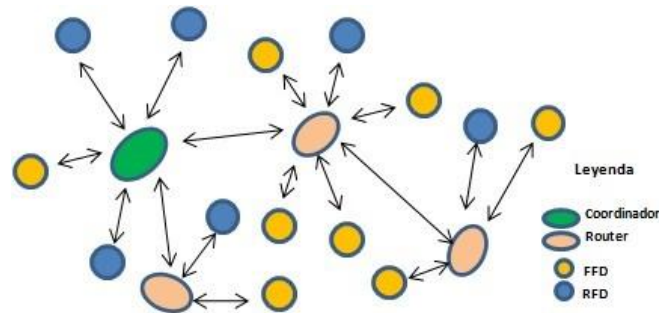


Figura 2.8.: Topología árbol o clúster [Mic13a].

- **Topología en malla:** Una red de malla es similar a una configuración de grupos de árboles, excepto que los FFDs pueden conmutar los mensajes directamente a otros FFDs en lugar de seguir la estructura de árbol como en la Figura 2.9. Las ventajas de esta topología son que la latencia de mensajes se puede reducir y la fiabilidad se incrementa.

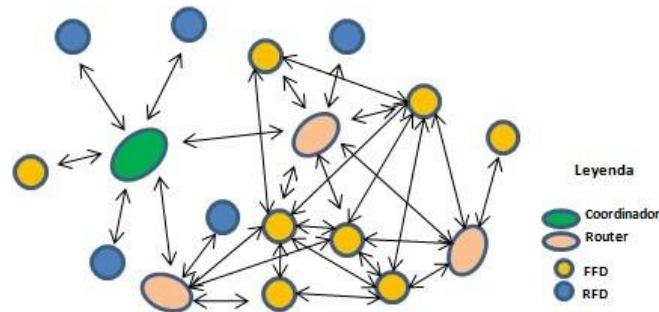


Figura 2.9.: Topología Malla [Mic13a].

2.2.6. Enrutamiento del árbol de ZigBee (ZTR)

ZTR está diseñado para dispositivos ZigBee con recursos limitados para elegir una ruta de enrutamiento de múltiple salto, sin algún procedimiento de descubrimiento de ruta, y funciona basado en un esquema de distribución jerárquica de direcciones como lo descrito en la Ecuación 2.1 y la Ecuación 2.2[TK14].

La Figura 2.10 muestra un ejemplo del esquema de asignación de direcciones en ZigBee, y la jerarquía de direcciones.

Cuando C_m (*mwkMaxChildren*), R_m (*nwkMaxRouters*), y L_m (*nwkMaxDepth*) son 3, 2, y 3 respectivamente. C_m , R_m , y L_m son definidos como el número máximo de hijos que puede tener un padre, el máximo número de routers que un padre puede tener como hijos, y el máximo nivel de árbol de una red en el estándar ZigBee, respectivamente.

En ZTR, cada nodo fuente e intermedio envía la información para cada nodo hijo descendiente, de otra manera, el envío de la información es a sus padres.

Mecanismo de asignación de direcciones de redes

El algoritmo utilizado para la asignación de direcciones se basa en la creación de subgrupos de direcciones. El coordinador siempre tiene asociada la dirección 0x0000. A cada uno de sus nodos hijos con capacidad de asignar direcciones, es decir routers, les asigna un conjunto de direcciones, asignando a este nodo router la primera de dicho conjunto y reservando las demás para posibles ampliaciones de la red. Una

$[Cskip[0] = 10, Cskip[1] = 4, Cskip[2] = 1]$

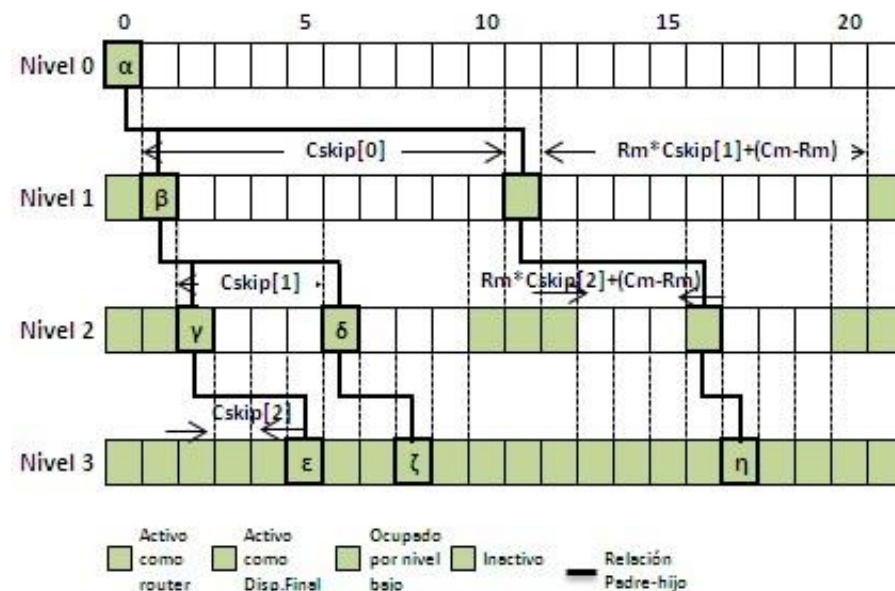


Figura 2.10.: Estructura de direcciones de ZigBee y relación Padre - Hijo (C_m 3, R_m 2, y L_m 3) [TK14].

vez que todos los routers hijos del coordinador tienen su conjunto de direcciones asignados, se les otorga las siguientes direcciones disponibles a cada dispositivo final hijo del coordinador. El mismo proceso realiza cada router con su conjunto de direcciones y los dispositivos asociados a él.

La relación entre nodo y nodo es la filiación en la red de clústeres de árboles. A cada nodo se le asignará una única dirección de red de 16 bits por su nodo padre en la red. El mecanismo de asignación se hace de la siguiente manera [JL10]: Cada nodo padre puede conectar C_m nodos hijos. Estos nodos pueden tener hasta R_m nodos de enrutamiento o

encaminamiento. La profundidad máxima de la red es $L_m.Cskip(d)$. Este es el número de direcciones que el nodo padre puede distribuir dada la condición de una profundidad de red y el número de nodos de enrutamiento y el nodos hijos. La profundidad de la red es d . El cálculo de $Cskip(d)$ es:

$$Cskip(d) = \begin{cases} 1 + C_m \cdot (L_m - d - 1) & R_m = 1 \\ -R_m - C_m \cdot R_m \cdot L_m - d - 1 & R_m \neq 1 \end{cases} \quad (2.1)$$

Cuando el valor $Cskip(d)$ de un nodo de enrutamiento es 0, éste no es capaz de distribuir direcciones para sus nodos hijos. Si el $Cskip(d)$ de un nodo de enrutamiento es mayor que 0, el nodo puede aceptar otros nodos como sus nodos hijos y distribuye direcciones de red para éstos. La dirección de un nodo que se une a la red primero, es uno más que el nodo padre. El intervalo de direcciones de otro nodo de enrutamiento es $Cskip(d)$. Cada nodo padre tiene R_m nodos hijos como máximo. La dirección de un nodo terminal es diferente a la del nodo de enrutamiento. Si la dirección del nodo padre es A_{parent} , la dirección del nodo hijo terminal es A_n .

$$\begin{aligned} A_k &= A_{parent} + Cskip(d) \cdot (k - 1) + 1 & 1 \leq k \leq R_m \\ A_n &= A_{parent} + Cskip(d) \cdot R_m + n & 1 \leq n \leq C_m - R_m \end{aligned} \quad (2.2)$$

Hay un ejemplo de asignación de direcciones de red en la Figura 2.11. Cuando $C_m = 4$, $R_m = 3$, $L_m = 4$.

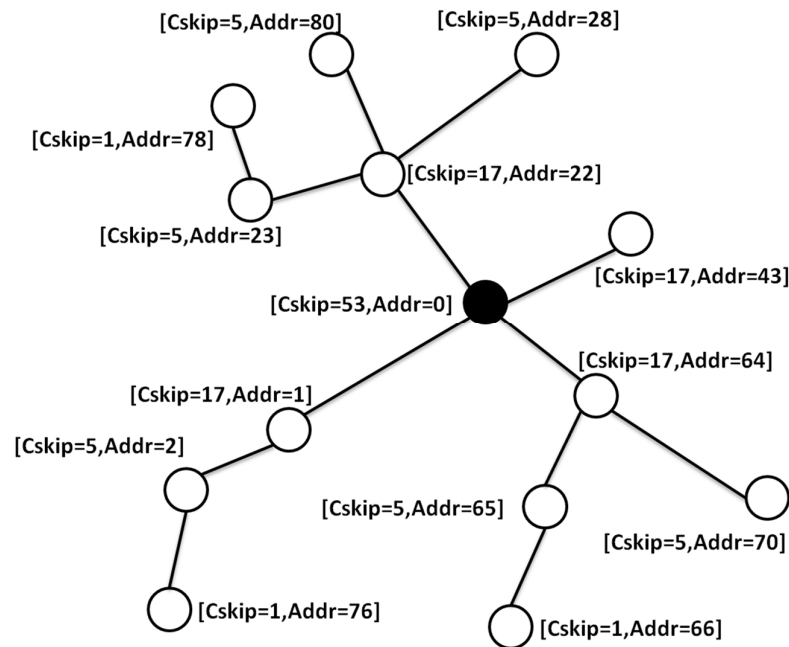


Figura 2.11.: Asignación de direcciones de red [Fuente: Elaboración Propia].

2.2.7. Tipos de Algoritmos para redes Cluster

Una red de simple cluster

Una cabecera de grupo (cluster head) y los nodos miembros, forman un grupo (cluster) y los grupos conforman una red. Esta sección describe la formación de un grupo de red simple y la comunicación dentro del grupo [EC01]. Una red de múltiples grupos son descritos a continuación.

Proceso de formación de cluster

Proceso de selección del jefe del grupo (cluster head). El proceso de formación de grupo comienza con la selección de una cabeza de grupo o jefe. Después que se selecciona una cabeza de grupo, éste expande enlaces con otros nodos miembros para formar un grupo.

En la Figura 2.12 se muestra que, después que un nodo se enciende, éste escucha y busca mensajes de saludo (HELLO) desde otros nodos. Si éste no puede obtener algún mensaje de saludo por cierto tiempo, entonces se vuelve una cabeza de grupo y envía mensajes de saludo a sus vecinos. La nueva cabeza de grupo espera respuestas desde

sus vecinos por un tiempo. Si ésta no recibe alguna solicitud de conexión, se vuelve en un nodo regular y empieza a escuchar de nuevo.

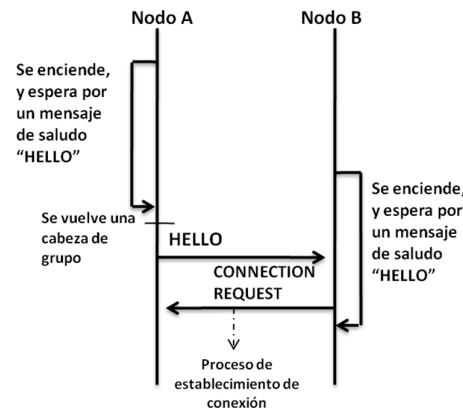


Figura 2.12.: Proceso de selección de cabecera de grupo [EC01].

Otros métodos para seleccionar la cabeza del grupo pueden ser posibles. La cabeza del grupo puede ser seleccionada basado sobre parámetros almacenados y calculados de cada nodo, como el rango de transmisión, la capacidad de potencia, la capacidad de computación o información de ubicación.

Cluster de simple salto

Después que un nodo es seleccionado como cabecera de grupo, éste difunde periódicamente un mensajes de saludo (HELLO) que contiene una parte de la dirección MAC en la cabecera del grupo y la ID 0 del nodo que indica la cabecera del grupo. Los nodos que reciben el mensaje de saludo envían un mensaje de solicitud de conexión (CONNECTION REQUEST) a la cabecera del grupo. Cuando la cabecera de grupo recibe la solicitud de conexión, éste responde al nodo con un mensaje de respuesta de conexión (CONNECTION RESPONSE) que contiene un identificador de nodo (ID). El ID debería ser único dentro de un grupo y la cabecera del grupo tiene la responsabilidad de asignar y gestionar un único ID para sus nodos miembros. Los nodos que tienen asignados un ID responden con un mensaje ACK a la cabecera de grupo. Una vez finalizada cada operación de intercambio de mensajes, ambos nodos se establecen como padres o hijos en su lista de vecinos. Específicamente, la cabecera del grupo establece como un nodo hijo a aquel nodo que se agrega a su lista de red y el nodo hijo establece la cabecera del grupo como padre. El enlace entre la cabecera del grupo y el nodo miembro se establece en este momento. La Figura 2.13 muestra lo antes mencionado.

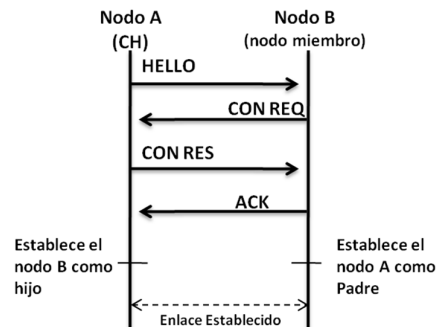


Figura 2.13.: Enlace configuración entre CH y nodo miembro [EC01].

Si todos los nodos son ubicados en el rango de la cabecera del grupo, (Figura 2.14), la topología de conexión llega a ser una estrella cada nodo miembro es conectado a la cabecera del grupo con un salto. El máximo número de nodos en un grupo es de 254 incluyendo la cabecera del grupo. El administrador o el fabricante pueden limitar las funciones del nodo para soportar solo un grupo de un solo un salto.

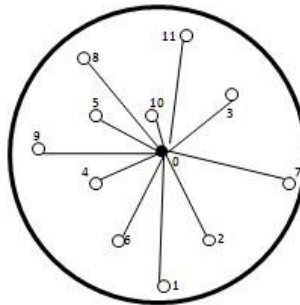


Figura 2.14.: Estructura de cluster de simple salto [EC01].

Cluster de múltiple salto

Un grupo puede expandirse a una estructura de múltiples saltos cuando cada nodo soporta múltiples conexiones. Aunque los retardos de la red se incrementan, la cobertura dentro un grupo puede aumentar.

El procedimiento de configuración de grupos de múltiples saltos es descrito en la Figura 2.15, Después que el nodo B ha establecido un enlace con la cabecera del grupo, este inicia una retransmisión del mensaje de saludo (HELLO) desde la cabecera del grupo. Cuando el nodo C obtiene el mensaje desde el nodo B, éste envía un mensaje de solicitud de conexión (CONNECTION REQUEST). El nodo B solicita un nuevo ID a la cabecera de grupo para asignarlo al nodo C. Cuando el nodo B recibe un nuevo ID desde la cabecera

del grupo, este envía un mensaje de respuesta de conexión (CONNECTION RESPONSE) al nodo B. Entonces el nodo C lo recibe y contesta con un mensaje ACK. Luego del intercambio de mensajes, el nodo C establece el nodo B como su padre, y el nodo B establece al nodo C como su hijo, y la cabecera del grupo establece al nodo C como nodo hijo de B. A continuación el nodo C inicia una retransmisión del mensaje HELLO para anunciarse a sus vecinos.

Cuando un nodo recibe varios mensajes de saludo desde diferentes nodos, éste responde al mensaje de saludo porque tiene una fuerte posibilidad de enrutamiento corto a la cabecera del grupo. El camino a la cabecera de grupo podría no ser ideal en este tiempo. La ruta de la cabecera del grupo se optimizará en un proceso posterior.

Este proceso de expansión puede continuar hasta que la cabecera del grupo quede sin todos los ID de los nodos. Además se puede limitar la máxima cantidad de saltos para reducir el retardo de la red.

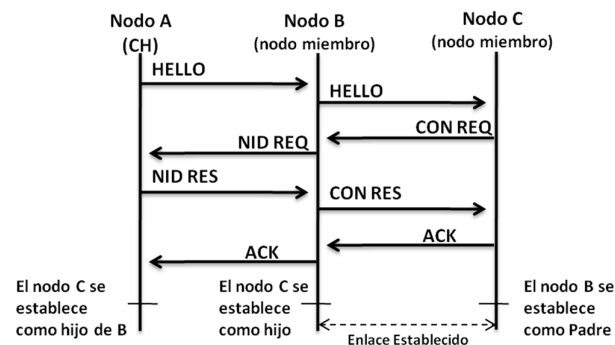


Figura 2.15.: Proceso de configuración de múltiple salto de grupo [EC01].

Rechazo de conexión

Cuando la cabecera del grupo ha agotado todos sus direccionamientos (IDs), o el grupo ha alcanzado otro límite definido, la cabecera de grupo debería rechazar las solicitudes de conexión de nuevos nodos. Para rechazar las solicitudes de conexión, el NID temporal, NID 254 es usado en el campo de destino NID del mensaje de respuesta de conexión (CONNECTION RESPONSE) o el nuevo campo NID o mensaje, respuesta de identificador del nodo (NODE ID RESPONSE).

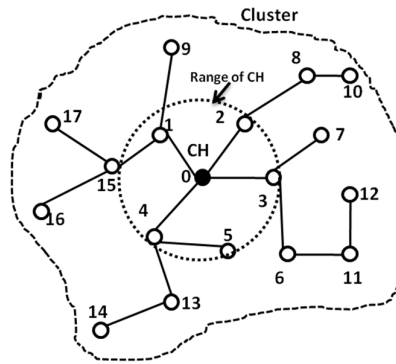


Figura 2.16.: Estructura de múltiple salto de grupo [EC01].

Cuando un nodo solicitante recibe un mensaje NODE ID RESPONSE con NID 254, este envía un mensaje de respuesta de conexión con NID 254 para el nuevo nodo. Si un nuevo nodo sigue recibiendo una CONNECTION RESPONSE con NID 254, almacena el grupo ID y detiene el envío de mensajes con CONNECTION REQUEST al nodo que pertenece al mismo grupo durante un tiempo.

Mantenimiento de red

Mensaje de saludo (HELLO) y lista de vecinos

En la Figura 2.17 la cabecera de grupo difunde periódicamente mensajes de saludo (HELLO) a sus nodos miembros. Cuando los nodos miembros reciben el mensaje de saludo desde la cabecera del grupo, ellos también envían un mensaje de saludo (HELLO) para anunciarse a sus vecinos. Cada nodo registra sus nodos vecinos en su lista de vecinos. La entrada de la lista de vecinos es actualizada por un mensaje de saludo periódico. Si una entrada de nodo no se actualiza en un cierto límite de tiempo de espera, este debería ser eliminado.

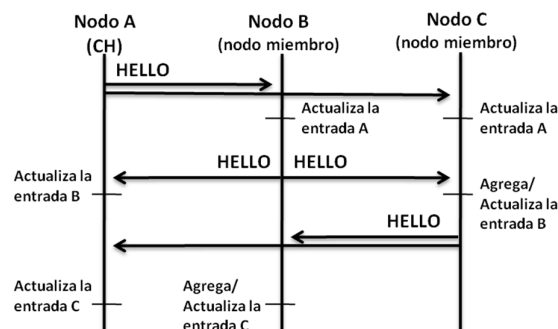
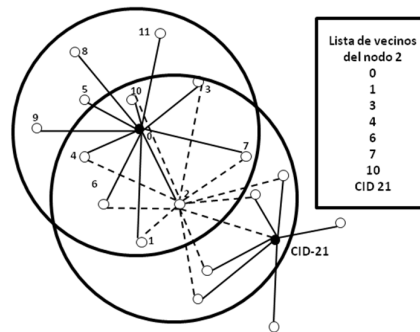


Figura 2.17.: Mensaje HELLO y la actualización de lista de vecinos [EC01].

Los nodos miembros pueden hablar directamente con los nodos vecinos. Si un nodo quiere comunicarse con otro nodo fuera de su rango, éste pregunta a la cabecera de grupo o al nodo padre para transmitir el mensaje a su destino.

Un nodo puede recibir un mensaje de saludo desde un nodo que pertenece a un grupo distinto. En este caso, el nodo agrega el ID del grupo (CID) del nodo de transmisión en la lista de vecinos.

**Figura 2.18.:** Lista de vecinos de un nodo en un grupo de borde [EC01].

Reporte de estado de enlace

En la Figura 2.19 cada nodo reporta sus estados de enlace a la cabecera del grupo. Un nodo miembro envía periódicamente un mensaje de reporte de estado de enlace (LINK-STATE REPORT) que contiene la lista de IDs de nodos vecinos para la cabecera del grupo. La frecuencia de mensajes de reporte de estado de enlace será determinado por requerimientos de aplicación y estabilidad.

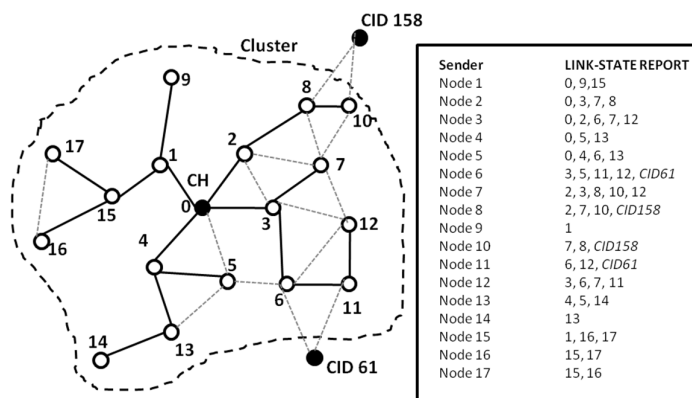


Figura 2.19.: Reporte de estado de enlace [EC01].

Actualización de topología

Basados en el mensaje del reporte de estado de enlace la cabecera de grupo periódicamente calcula los caminos cortos entre él y sus nodos miembros e informa esto a los miembros por un mensaje de actualización de topología (TOPOLOGY UPDATE).

La regla de selección de ruta es sencilla. La cabecera del grupo debería escoger la ruta con menor cantidad de saltos. Si hay varias rutas con la misma cantidad de saltos, la cabecera de grupo debería elegir la ruta que tiene el ID de nodo más pequeño que el nodo padre o alguna regla arbitraria similar.

En la Figura 2.20, se observa que, si un nodo miembro recibe el mensaje de actualización de topología diferente que el nodo padre, se cambia el nodo padre como se indica en el mensaje. Y también se registra sus nodos hijos y los nodos debajo de él en el árbol en ese momento. Los nodos dentro de un grupo se comunican básicamente con otros nodos a través del nodo padre, excepto el caso donde se comunican con sus nodos vecinos directamente. El ciclo de la actualización de la topología depende sobre el ciclo de reporte de actualización del enlace.

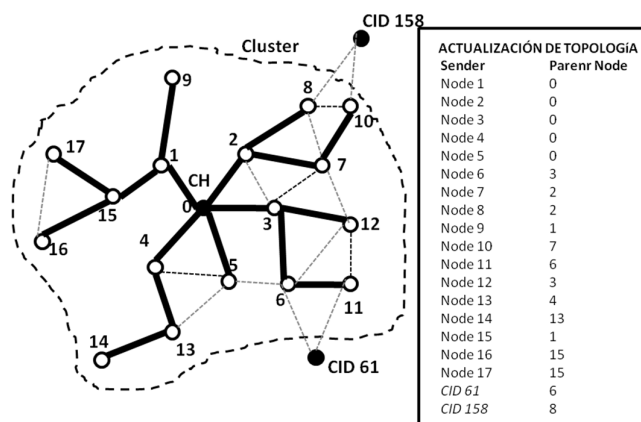


Figura 2.20.: Actualización de topología [EC01].

2.3. Filtros de Bloom

Los filtros de Bloom se introdujeron en 1970 por Burton Howard Bloom, cuando la memoria del ordenador era muy escasa [Blo70b]. Las primeras aplicaciones incluyen la separación de sílabas, diccionarios ortográficos, y tablas de unión en las bases de datos. Con la llegada de Internet, su utilidad fue redescubierta. Un Filtro de Bloom (BF), se utiliza para comprobar si

2.3 Filtros de Bloom

un elemento es miembro de un conjunto. La coincidencia de falsos positivos son posibles, pero no son los falsos negativos.

Los elementos pueden ser añadidos al conjunto, pero no se eliminan. Si es mayor los elementos que se añaden al conjunto, mayor será la probabilidad de falsos positivos.

Un filtro de Bloom es una estructura de datos que tiene por objetivo compactar un conjunto de datos, es decir; utiliza un vector de bits para representar un conjunto de datos. Para que ocurra esto hace uso de funciones hash.

Éste vector de bits cuenta inicialmente con valor de cero. El filtro admite la utilización de dos algoritmos o la inserción de un elemento en el conjunto representado por el filtro y la prueba de pertenencia que verifica si un elemento pertenece o no a un conjunto representado por el filtro. Caso que el algoritmo insertado se halla utilizado anteriormente la prueba tendrá una respuesta positiva, sin embargo el filtro de Bloom tiene una desventaja que es la posibilidad de ocurrencia de falsos positivos, es decir; que se considera que un elemento externo pertenezca a el conjunto de elementos insertados en el filtro y la tasa de ocurrencia de falsos positivos aumenta conforme se eleva el número de elementos insertados en el filtro, el filtro de Bloom posee una ventaja significativa que es la economía de espacio.

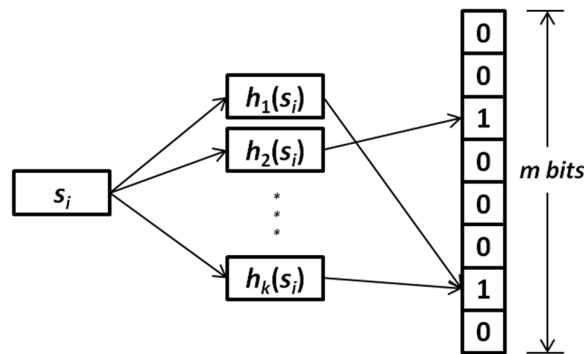


Figura 2.21.: Inserción de un elemento en un filtro de Bloom [RPL05].

La formula siguiente indica la probabilidad de un determinado bit permanezca en cero después de insertados los n elementos, o sea la probabilidad que ocurra un falso positivo[Blo70b].

$$p = 1 - \left(1 - \frac{1}{m}\right)^{kn}$$

2.3.1. Funcionamiento de los filtros de Bloom

El funcionamiento de un BF comprende dos métodos: (1) inserción de elementos para almacenar nuevos elementos en el filtro; y (2) consulta de pertenencia (contención) para verificar si un elemento está presente en el filtro. Es necesario destacar en este punto que los BF no proporcionan soporte para la eliminación de entradas por diseño.

Inserción de elementos:

Un BF es típicamente implementado como un arreglo de m bits, que son inicialmente configurados con 0. Con el fin de insertar un elemento de datos en el arreglo de bits, las h funciones hash son calculadas sobre el elemento a insertar. En nuestro caso, un elemento de este tipo es una dirección única de un nodo sensor. Cada una de las funciones hash devuelve el índice de un elemento del arreglo de bits, que es puesto en 1. La inserción de un elemento de este modo corresponde al ajuste de los h bits señalados por las funciones de hash a 1. En caso de que un bit ya ha sido puesto en 1, antes de la operación de inserción, su estado no se altera. En la Figura 2.22 se muestra la inserción de dos elementos, *input1* e *input2*, en el BF con $m = 16$ bits y $h = 3$. La figura también muestra una colisión en la sexta posición del BF, donde las funciones de hash asignan la misma entrada para ambas secuencias de entrada. Estas colisiones causan la aparición de falsos positivos, como se explica a continuación.

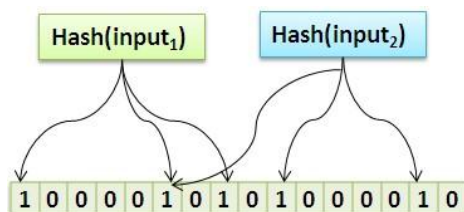


Figura 2.22.: Inserción de dos elemento en el filtro de Bloom. ($m=16$, $h=3$) [AR12].

Consulta de pertenencia:

Con el fin de comprobar si un elemento está contenido en el BF, las funciones hash del elemento de entrada son calculados primeramente. Si cualquiera de los bits en las posiciones h en el BF está puesto en 0, el elemento de entrada, sin duda, no figura en el BF. Cuando todos los bits en las posiciones h se ponen a 1, el símbolo de entrada es un miembro del BF con una cierta probabilidad. Dependiendo del número de funciones hash y la longitud del arreglo de bits, una consulta de pertenencia puede sin embargo devolver una declaración de falsos positivos.

La probabilidad de falsos positivos depende del valor de m que denota el tamaño del BF y h el número de elementos insertados en el filtro, así como en el número de funciones hash t

insertados en el BF. Bajo la suposición de que las salidas de las funciones hash son independientes y distribuidas de manera uniforme, la probabilidad de una consulta de pertenencia para volver un falso positivo, indicado como P_{FP} , se calcula como [Blo70a]:

$$P_{FP} = \left(1 - \left(1 - \frac{1}{m}\right)^h\right)^m \quad (2.3)$$

Nótese que para valores más grandes de m la P_{FP} disminuye, pero con un conjunto mayor de elementos a ser insertados (t), se incrementa. La estimación específica para la probabilidad de falsos positivos que desee y para el número de elementos esperados (t) permiten especificar los parámetros de un BF como $h = \lceil -\log_2 P_{FP} \rceil$ y $m = \lceil \frac{ht}{\ln(2)} \rceil$.

Ejemplo de uso de filtros de Bloom

Un filtro de Bloom se construye sobre la base de un simple algoritmo utilizado para determinar la presencia de una función basada en los datos. Un filtro de Bloom permite falsos positivos, pero el ahorro de espacio a menudo supera este inconveniente, cuando la probabilidad de un error está bien controlada. Un filtro de Bloom es un arreglo de bits y se compone de funciones hash. El siguiente ejemplo muestra cómo un filtro Bloom se comporta en una búsqueda de secuencias [TO08].

1. Para generar un filtro Bloom, utilizamos una matriz de m bits y usamos h funciones hash independientes. Inicialmente todos los bits se ponen en 0, mientras que un valor hash puede tomar cualquier valor de 1 a m . Como se muestra en la Figura 2.23., se utiliza una matriz de 10 bits y 3 funciones hash.
2. Entonces asumimos "5126 4353 8934", como los datos de destino, que se dividen en las 3 claves de "5126", "4353" y "8934". Cada clave es ordenada por cada función hash para producir una ubicación de bits, los cuales se ponen en 1. En la Figura 2.23, "5126" es ordenado a 1, 7 y 9 por las funciones hash y cada uno de los bits correspondientes se establece en 1.
3. Para generar un filtro Bloom para "5126 4353 8934", ejecutamos una operación lógica OR entre las matrices de bits.
4. Para buscar una cadena de "4353" desde los datos, la clave hash de las 3 funciones hash generan una matriz de bits. Luego comparamos el conjunto de bits con el filtro Bloom. Si todos los bits están puestos a 1 en el conjunto de los bits, también se ponen a 1 en el filtro Bloom. Hay una posibilidad de que la cadena de "5126 4353 8934" se incluya en los datos. De lo contrario, "4353", sin duda no se incluye en los datos.

Un filtro de Bloom puede producir un falso positivo, donde se sugiere que una clave puede estar en los datos a pesar de que no está realmente incluido en los datos. Debido a que el

resultado de la operación OR en (3) podría causar falsos positivos. Pero un filtro de Bloom permite una posibilidad de falsos positivos, siempre y cuando la probabilidad sea pequeña.

Información Propuesta = "5126 4353 8934"

Claves = "5126", "4353", "8934"

$$\left\{ \begin{array}{l} \text{Hash 1(5126)} = 1 \\ \text{Hash 2(5126)} = 7 \\ \text{Hash 3(5126)} = 9 \end{array} \right. \quad \left\{ \begin{array}{l} \text{Hash 1(4353)} = 6 \\ \text{Hash 2(4353)} = 10 \\ \text{Hash 3(4353)} = 3 \end{array} \right. \quad \left\{ \begin{array}{l} \text{Hash 1(8934)} = 3 \\ \text{Hash 2(8934)} = 6 \\ \text{Hash 3(8934)} = 2 \end{array} \right.$$

5126	1	0	0	0	0	1	0	1	0
4353	0	0	1	0	0	1	0	0	1
8934	0	1	1	0	0	1	0	0	0
5126 4353 8934	1	1	1	0	0	1	1	1	1
Búsqueda de Clave = "4353"									
4353	0	0	1	0	0	1	0	0	1

Figura 2.23.: Ejemplo de filtro de Bloom[TO08].

3. Estado del Arte

En la Fig.3.1 mostramos el mapa conceptual de la literatura referida a técnicas de enrutamiento en redes de sensores y otros tipos de redes.

3.1. Enrutamiento en Redes de Sensores Inalámbricos (WSN) / Capa de Red

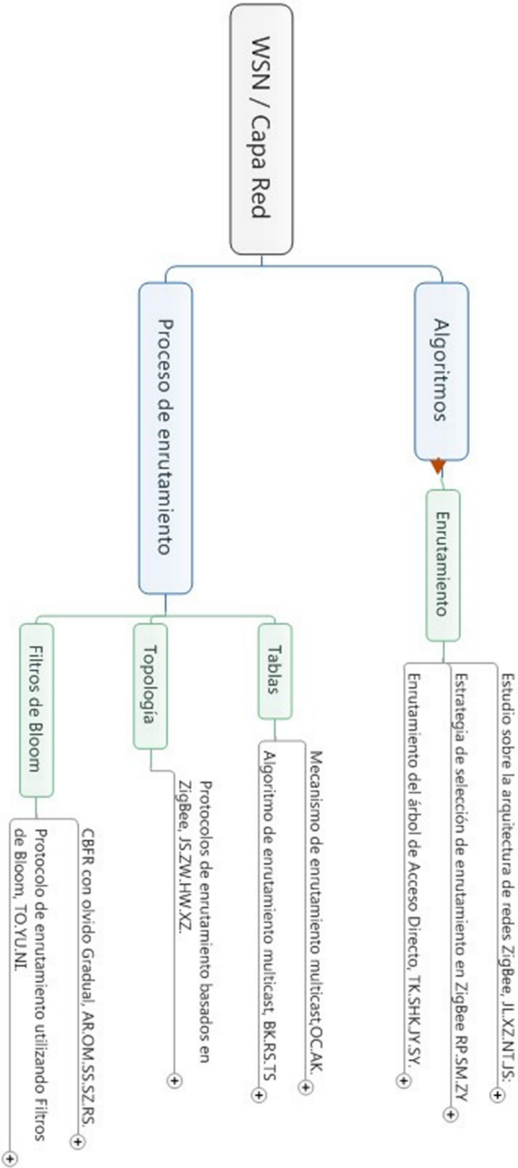
3.1.1. Algoritmos de Enrutamiento en ZigBee

El objetivo propuesto por los autores en [JL10], es reducir considerablemente la sobrecarga en el control de la red. La solución de enrutamiento ZigBee (ZBR) es una propuesta basado en el algoritmo de enrutamiento ZigBee. Los resultados generados fueron desarrollados en el simulador NS-2, el cual les permitió realizar diversas pruebas en una misma plataforma, lo cual podría considerarse ideal para redes de sensores inalámbricos (WSN). Las soluciones de mejora son consideradas desde dos perspectivas. Una es controlando la distancia máxima ZBR y colocando un control de sobrecarga. La otra es limitando la dirección de cada paquete y colocando la sobrecarga en dirección opuesta del control del mismo. El control de sobrecarga se reduce, obviamente, con la solución ZBR mejorada. Los resultados de las simulaciones indican que la sobrecarga por control es reducida sin diezmar la entrega de paquetes y en la longitud de la trayectoria de las rutas.

El protocolo de enrutamiento juega un rol importante en el desempeño de las redes. La sobrecarga de control es un indicador muy significativo para medir el desempeño del protocolo de enrutamiento. Si incrementamos la sobrecarga de control, éste consumiría más energía de los nodos. Los autores con el algoritmo propuesto han permitido tener un balanceo del consumo de energía de las redes, permitiendo extender la vida de la red, pero no llegan a considerar la importancia de integrar en sus redes una comunicación multicast.

En [RP06], los autores introducen la combinación de estrategias de enrutamiento de AODV (Ad hoc On Demand Distance Vector) y enrutamiento árbol de ZigBee. Una estrategia de selección de enrutamiento basados en servicio de datos y un algoritmo de balance de energía de ZigBee son propuestos. Realizaron varias simulaciones utilizando NS-2 para evaluar ambos algoritmos. La mejora permite tener un mejor control de la energía usada en los dispositivos. Consideramos de gran importancia el

Figure3.1.: Mapa conceptual de la técnica de enrutamiento en WSN y otros tipos de redes [Fuente: Elaboración Propia].



aporte de estos autores referente al análisis del algoritmo de enrutamiento árbol, ya que se refiere a una jerarquía, ya pre-establecida, con sus grados de profundidad.

Análisis del algoritmo de enrutamiento - Enrutamiento árbol

El mecanismo de enrutamiento árbol incluye la configuración de las direcciones y el enrutamiento árbol. Cuando el coordinador establece una nueva red, éste se asigna una dirección 0 y una profundidad de red $Depth0 = 0$. Si el nodo(i) necesita unirse a la red, asociando al nodo(k), el nodo(k) se convertirá en un nodo padre del nodo (i). Entonces, de acuerdo a su propia dirección de red A_k y profundidad de red $Depthk$, el nodo (k) asignará una dirección de red A_i y una profundidad de red de $Depthi = Depthk + 1$ para el nodo (i). La profundidad de la red representa el número de saltos desde un nodo al nodo raíz; es decir, el coordinador de la red.

La Fig.3.2 muestra la estructura árbol de ZigBee. El parámetro $nwkMaxChildren$ (CM) representa el máximo número de hijos que pueden asociarse a un router o un coordinador. El parámetro $nwkMaxRouter$ (RM) es el número máximo de nodos hijos que pueden ser router.

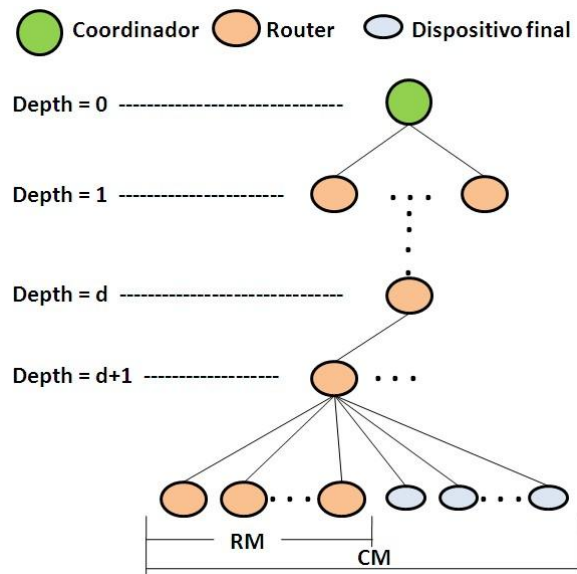


Figura 3.2.: Estructura árbol de ZigBee [RP06].

Un nuevo nodo (n) es un RFD (dispositivo de función limitada), cuando no tiene capacidad de enrutamiento. Éste es asociado a un coordinador como su nodo hijo. De acuerdo a su propia profundidad d , el nodo padre k asigna una dirección de red al nodo hijo n :

$$A_n = A_k + Cskip(d).Rm + n \quad (3.1)$$

3.1 Enrutamiento en Redes de Sensores Inalámbricos (WSN) / Capa de Red

Y si el nuevo nodo hijo es un FFD (dispositivo de función completa), éste tiene capacidad de enrutamiento. El nodo padre k le asignará también una dirección de red $A_n = A_k + Cskip(d).(n - 1)$, y un $Cskip$ igual a:

$$Cskip = \begin{cases} 1 + Cm.(Lm - d - 1), & \text{si } Rm = 1 \\ \frac{1+Cm-Rm-Cm.Rm.Lm-d-1}{1-Rm}, & \text{en caso contrario} \end{cases} \quad (3.2)$$

Se supone que un router envía un paquete de datos al nodo de destino cuya dirección de red es D . La dirección de red y la profundidad de la red del router son A y d respectivamente, Si el nodo destino es su nodo hijo: $A < D < A + Cskip(d-1)$, y la dirección del nodo del siguiente salto es:

$$N = \begin{cases} D, & \text{si es dispositivo final} \\ A + 1 + \frac{[D-(A+1)]}{Cskip(d)} Cskip(d), & \text{de otra manera} \end{cases} \quad (3.3)$$

De otra manera, el siguiente salto es el nodo padre de su router.

En [TK14] los autores identificaron que en el algoritmo ZTR (ZigBee Tree Routing), que es el algoritmo de enrutamiento-árbol de ZigBee, se tiene el problema de desvío de ruta (el cual consiste en tener que pasar siempre por el coordinador para el envío de paquetes) y el problema de concentración de tráfico. Para dar solución a estas dificultades, los autores proponen un algoritmo que permite alcanzar a otros nodos sin realizar tantos saltos, llamado STR (Shortcut Tree Routing), o enrutamientoárbol de acceso directo, el cual usa las tablas de nodos vecinos, que son típicamente definidas en el estándar ZigBee cuando construye su red árbol mediante un proceso de asignaciones de direcciones por nivel.

En STR, cada nodo puede encontrar el salto óptimo al siguiente nodo, basándose en los saltos restantes del árbol hacia un destino (ver Fig.3.3). Los análisis matemáticos muestran que usando la información de un vecino antecesor ayuda a tener comunicaciones con un menor número de saltos (y así proporcionar un camino de enrutamiento eficiente), con lo cual es posible reducir el tráfico de carga concentrada en los enlaces del árbol.

El algoritmo STR, básicamente, escoge uno de los nodos vecinos como el siguiente salto del nodo, lo cual permite reducir el número de saltos restantes hacia el nodo destino. Por ejemplo en la Fig.3.3, STR calcula los saltos restantes siguiendo el árbol hacia el nodo destino $D2$, y compara dicho número con otra ruta que pase por un nodo vecino.

Finalmente, selecciona el *N4* como el siguiente nodo al cual saltar para luego transmitir el paquete para el destino *D2*.

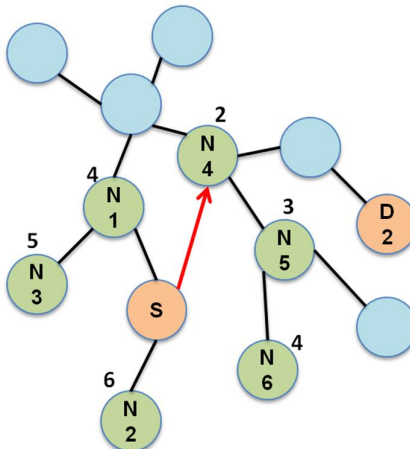


Figure 3.3.: Enrutamiento-Árbol de Acceso Directo (STR) [TK14].

3.1.2. Proceso de enrutamiento

Tablas

El estándar ZigBee no considera el enrutamiento de multidifusión (multicast) en su especificación. Los autores en [OG10] han propuesto el mecanismo de enrutamiento multicast Z-Cast, para la comunicación entre los nodos sensores que pertenecen al mismo grupo. El mecanismo Z-Cast garantiza que un mensaje multicast enviado desde un miembro del grupo llega a todos los demás miembros del grupo, al tiempo que reduce el número de paquetes transmitidos. Esto se logra descartando los mensajes en las hojas que no contienen a miembros del grupo. También han demostrado cómo integrar el mecanismo de Z-Cast en la pila de protocolos IEEE 802.15.4/ZigBee con complementos menores. Además demuestran que el mecanismo propuesto es eficiente al minimizar considerablemente el número de mensajes transmitidos entre los miembros del grupo.

Muchas de las propuestas de enrutamiento multicast para redes ad-hoc se han tratado de aplicar a las WSN, pero dichas soluciones han sido diseñadas para trabajar con nodos con gran capacidad de almacenamiento y alto poder computacional. Parte de la solución que proponen estos autores [OG10], es la creación de nuevas tablas de enrutamiento, que sólo están disponibles en el nodo coordinador y en los nodos intermedios (routers). En los nodos finales no es necesario tener dichas tablas pues no enrutan ningún mensaje. Las actualizaciones de las tablas se dan de la misma forma; al formar un nuevo grupo, éste también debe ser agregado a la tabla de enrutamiento de cada nodo, sea coordinador o router. Los autores vienen realizando más estudios para lograr obtener

una implementación mas real y válida para el estándar ZigBee, realizando pruebas para verificar su efectividad.

En [BK07] los autores describen el desarrollo de M-Bee, un algoritmo de enrutamiento multicast basado en los estándares IEEE 802.15.4 y Zigbee. Hay que tomar en cuenta que el estándar ZigBee solo admite transmisiones unicast y broadcast, el concepto multicast no es muy aceptable actualmente para ZigBee. Los autores refieren que, si se introduce la difusión multicast, esta sólo se utilizará para establecer una ruta (o árbol) multicast, y luego todos los paquetes serán transmitidos a través de ese árbol, lo cual reduce la cantidad de tráfico.

El modelo multicast propuesto está basado en dos procedimientos principales: descubrimiento de rutas (RD), y la unión de receptores (RJ). Durante la fase de RD, un nodo emisor intenta localizar los receptores interesados en una información en especial, y forma una ruta entre ellos. Esta ruta se ve reforzada durante la fase de RJ, en la que cada receptor declara su adhesión a la ruta a la fuente de datos. Por lo tanto, la difusión broadcast es reducida y el tráfico de red se ve disminuido. En dicho trabajo, se midió el tiempo de respuesta desde que se inicia el periodo de descubrimiento hasta la unión de los receptores. A manera de sugerencia, mencionamos que hubiera sido bueno también evaluar cuántos paquetes se enviaron luego de haberse establecido la red, cuántos han llegado a su destino final y cuántos fueron descartados. Un hecho anecdótico, es que los autores encontraron algunas inconsistencias en los esquemas de direccionamiento de la pila microchip.

Topología

Los autores en [JS], hacen una revisión de tres esquemas de enrutamiento ZigBee: AODV (y AODVjr), Cluster-tree, y enrutamiento integrado. Adicionalmente, describen las fórmulas para analizar los mecanismos de enrutamiento y los costos correspondientes. Finalmente, introducen un enfoque para reparar la red cuando algún dispositivo falla.

AODV y AODVjr son un algoritmo de adquisición de enrutamiento bajo demanda. El nodo origen difunde un paquete de solicitud de ruta (RREQ) a sus vecinos (que suelen ser nodos de enrutamiento), los cuales lo reciben y a su vez lo re-difunden a sus vecinos hasta que lleguen al nodo final. Durante el proceso de re-difusión de la solicitud de RREQ, los nodos intermedios registran la dirección del nodo transmisor, desde el primer paquete de difusión que fue recibido en sus tablas de descubrimiento de rutas, por lo tanto se establece una ruta de regreso. La comparación de los costos de ruta de los paquetes con el mismo RREQ permite elegir el mejor camino en ese momento y descartar el peor. Una vez que el nodo destino recibe la RREQ, este responde por unicast una respuesta de ruta (RREP) de regreso a sus vecinos desde el cual recibió la RREQ.

El AODVjr es una de las versiones más simplificadas de AODV. AODVjr retira de la especificación AODV, la secuencia de números, RREP, cantidad de saltos, mensajes de

saludo (HELLO) y lista de precursores. En AODVjr, si las comunicaciones son unidireccionales, el destino manda un mensaje de conexión al emisor. Si el tráfico de la información es bidireccional, no se utilizan mensajes adicionales. Un emisor detecta un enlace roto en una ruta cuando éste no recibe un mensaje desde el destino.

El algoritmo Cluster-tree es un protocolo de enlace lógico y de capa de red que usa paquetes de estado del enlace para formar, ya sea una red de grupos individuales, o una red de grupos de árboles. Las redes básicamente se reorganizan y soportan redundancia (para alcanzar un grado de resistencia), en caso de fallo y autorreparación. Los nodos seleccionan a un nodo como cabecera de grupo (Cluster Head, CH), y forman un grupo, conforme al modo auto-organizado. Los grupos se interconectan usando unos nodos denominados Dispositivos Designados (DD).

Tal como lo indican los autores, una red de simple cluster contiene una cabecera de grupo (CH), el cual difunde periódicamente mensajes de saludo (HELLO), los cuales contienen una parte de la dirección MAC del CH y el ID=0 (que indica que el nodo es la cabecera de grupo). Los nodos que reciben este mensaje envían un mensaje de solicitud de conexión (CONNECTION REQUEST) a la cabecera del grupo (CH). Cuando el nodo CH recibe el mensaje, responde con un mensaje de respuesta de conexión (CONNECTION RESPONSE), el cual contiene un ID para el nodo que solicitó la conexión (el ID del nodo corresponde a una dirección corta en la capa MAC). El nodo que es asignado con el ID responde con un mensaje ACK a la cabecera del grupo. Si todos los nodos son ubicados en el rango de cobertura de la cabecera del grupo, la topología de conexión llega a ser una estrella y cada nodo miembro es conectado a la cabecera del grupo con un solo salto. Un cluster puede extenderse dentro de una estructura multi-salto cuando los nodos tienen capacidad de soportar múltiples conexiones (nodos router intermedios).

Para una red conformada por múltiples grupos, es necesario contar con dispositivos designados (DD). Si un miembro viene recibiendo mensajes de saludo (HELLO) desde un DD, éste agrega un CID=0 (que es el ID del DD) en su lista de vecinos y reporta al CH. El reporte hacia el CH selecciona el nodo miembro como un nodo de borde (es decir, el dispositivo designado) para su cluster padre y envía un mensaje de solicitud de conexión de red (NETWORK CONNECTION REQUEST) para el nodo miembro junto a una conexión con el DD. El nodo de borde solicita una conexión y une al cluster 0 como su nodo miembro, entonces envía un mensaje solicitando el CID (CID REQUEST) al DD. Cuando el CH obtiene el nuevo CID, éste informa a sus nodos miembros con un mensaje de saludo (HELLO).

En el algoritmo de enrutamiento integrado, un nodo puede ser de una de las dos clases siguientes: nodo de enrutamiento mas (RN+), que tiene memoria suficiente para enrutar según AODVjr, o nodo de enrutamiento menos (RN-), que tiene un límite de memoria y sólo realiza enrutamiento cluster-tree. Mientras un nodo RNsiempre sigue un árbol cluster, un nodo RN+, o bien puede seguir el árbol del cluster, o puede descubrir dinámicamente una ruta AODV, dependiendo de varios factores, como por ejemplo la duración de la sesión, la tolerancia al retardo de descubrimiento de ruta, etc.

3.1 Enrutamiento en Redes de Sensores Inalámbricos (WSN) / Capa de Red

El enrutamiento cluster-tree favorece a dispositivos con limitaciones de memoria y

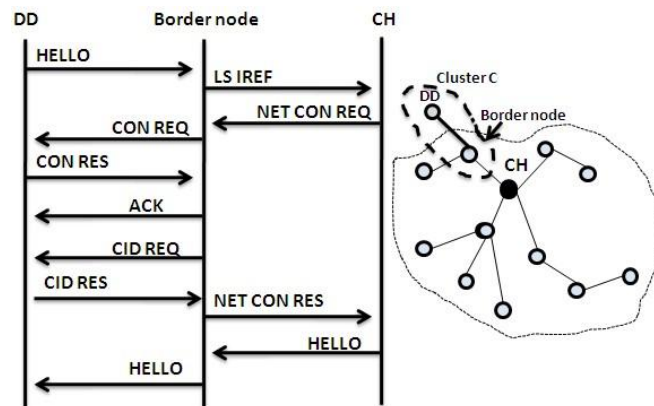


Figura 3.4.: Asignación del CID [JS].

es muy adecuado para sesiones cortas de comunicación. La mayoría de rutas de árbol, en términos de cantidad de saltos, no son óptimas. AODV y AODVjr son capaces de encontrar rutas óptimas o casi óptimas, y ayudan a reducir la latencia de mensajes entregados. En general, AODV y AODVjr son adecuados para los dispositivos con memoria insuficiente, y favorecen las sesiones con muchas comunicaciones. El enrutamiento integrado combina estas dos rutas y hace equilibrio entre ellas de acuerdo a las condiciones y requisitos de la red.

Cuando un enlace o dispositivo falla, los dispositivos inician una reparación de ruta. Si el dispositivo superior (coordinador, o cabecera) no es capaz de iniciar la reparación de ruta debido a la falta de capacidad de enrutamiento o alguna otra limitación, éste deberá emitir un comando de error de ruta al dispositivo fuente, con el código de error que indica el motivo del error. Si un dispositivo final RFD (dispositivo de función limitada) no es capaz de transmitir mensajes a su padre, deberá iniciar el procedimiento de orfandad, que permite al dispositivo desactivarse un periodo de tiempo, luego el dispositivo se activa y realiza una nueva búsqueda de su nodo padre. Si el procedimiento de orfandad tiene éxito y el dispositivo final se re-establece con su padre, el dispositivo final deberá reanudar el funcionamiento en la red. Si el procedimiento de orfandad falla, el dispositivo final intentará volver a conectarse a la red a través de un nuevo padre. De igual modo en una red árbol, cuando se pierde sincronización con los dispositivos descendentes con su padre, o es incapaz para transmitir un mensaje a su padre, el dispositivo puede iniciar un proceso de orfandad para buscar a su padre o un proceso de asociación para encontrar un nuevo padre.

Enrutamiento basado en filtros de Bloom

El enrutamiento ZigBee está centrado en un solo nodo (llamado nodo coordinador), el cual se encarga de la asignación de nuevas direcciones a los distintos nodos (sean nodos intermedios o dispositivos finales) que quieren ser parte de su red. Además, a través de actualizaciones es capaz de construir su propia topología árbol que permite establecer comunicación con todos los dispositivos participan en su red. De esta manera, todo el tráfico generado en el enrutamiento ZigBee está concentrado en un solo nodo.

Como se recuerda, el enrutamiento clásico en todos los tipos de redes, está basado en tablas de enrutamiento que están almacenadas en cada uno de los nodos de la red. Una tabla de enrutamiento alberga la información de reenvío de los paquetes; consiste en una lista de direcciones de destino asociadas a cada uno de los puertos de salida. Dado que la cabecera de cada paquete guarda la información de origen y de destino, cuando el paquete llega a un nodo, se verifica en la tabla de enrutamiento hacia dónde se debe reenviar dicho paquete.

En el caso de ZigBee, estas tablas de enrutamiento son construidas cuando el nodo coordinador asigna las direcciones y forma las rutas para alcanzar a todos sus nodos miembros. Aquí es necesario que toda comunicación pase por el nodo coordinador, el cual alberga una tabla de enrutamiento, al igual que los nodos intermedios, los cuales permiten enrutar a su destino final paquete enviado.

Últimamente en diversas investigaciones se ha propuesto un cambio de paradigma en el enrutamiento de paquetes de datos. Dado que, es posible que las tablas de enrutamiento en los nodos de una red crezcan de manera exponencial a medida que se tienen más rutas o árboles, en este nuevo paradigma lo que se busca es reducir o desaparecer el uso de estas tablas.

Pero ¿cómo sería esto posible, si no se guardara la información del reenvío en los nodos intermedios de la red? La respuesta es: codificando la ruta que el paquete debe seguir, en la misma cabecera del paquete. Para lograr esta codificación, se han propuesto diversas técnicas [GM12], [AR12], [dJGD10], [TO08], [Her13], pero últimamente se está tomando como las más efectivas, las que están basadas en filtros de Bloom. Un filtro de Bloom es una estructura de datos que tiene por objetivo compactar un conjunto de datos, es decir, utiliza un vector de bits para representar un conjunto de datos. Para que ocurra esto hace uso de funciones hash, las cuales funcionan como un codificador y nos devuelven elementos que se pueden almacenar en dicho vector [Ver sec.2.3].

Cuando se usa filtros de Bloom, en lugar de especificar una ruta a través de tablas de enrutamiento, toda ruta se resume y se codifica en la cabecera de los paquetes. Cuando se envía el paquete, éste sólo consulta a través de operaciones AND si éste es el camino y el nodo por el cual pasar y/o entregar el mensaje. Esto significa justamente el cambio completo del paradigma de enrutamiento típico usado por ZigBee.

Los autores en [AZN10] proponen la técnica de enrutamiento multicast MPSS (MultiProtocol Stateless Switching), para un escenario de redes MPLS. En este trabajo se

encaminan los paquetes a través de filtros Bloom, reduciendo la información de estado en los nodos. Así, el mecanismo de conmutación de etiquetas MPLS es reemplazado por el mecanismo de reenvío basado en filtros Bloom. En MPSS, el funcionamiento básico basado en filtros de Bloom (ver Fig.3.5), consiste en que cada enlace unidireccional es identificado por un conjunto de m bits representando un filtro de Bloom, de los cuales k bits son 1, teniendo en cuenta que $k \ll m$. Estos identificadores de enlaces son generados aleatoriamente por cada nodo. Una vez que la ruta/árbol es calculada, para construir el filtro de Bloom para el enrutamiento de origen, los identificadores de enlace que forman el camino/árbol son operados conjuntamente en una operación OR. El arreglo resultante es el filtro de Bloom de reenvío, conocido como zFilter, el cual reemplaza la etiqueta MPLS en el paquete. Los nodos intermedios mantienen sólo una pequeña tabla de identificadores de enlace de sus propias interfaces de salida, eliminando así, casi por completo, las tablas de enrutamiento. Los enrutadores verifican los paquetes entrantes con las entradas de la tabla haciendo una operación AND del zFilter con los identificadores de enlace. Si el resultado es el identificador mismo, se permite que el paquete pase, si no, se descarta.

En MPSS el paquete usa un sólo filtro de Bloom para todo su enrutamiento. A medida que la ruta/árbol es más larga, debido a la operación OR de todos los enlaces participantes de la ruta/árbol, el filtro de Bloom se llena de más bits puestos en 1. Esto aumenta la posibilidad de la aparición de falsos positivos, lo cual provocaría varias anomalías de reenvío: duplicación de paquetes, bucles e inundación de paquetes espúreos. Para reducir este impacto negativo, una solución podría ser utilizar filtros mas grandes, sin embargo esto incrementaría también el ancho de banda desperdiciado por el uso de cabeceras de gran tamaño.

Variantes del enrutamiento basadas en filtros de Bloom

Dado que el enrutamiento con filtros de Bloom implica asumir el riesgo de la aparición de falsos positivos y de una serie de anomalías del reenvío, posteriormente algunos autores han propuesto unas variantes a esta técnica, con el fin de reducir estas anomalías. En un escenario de redes multicast MPLS (Multiprotocol Label Switching) basadas en VPNs, en [GM12] los autores proponen un nuevo método de enrutamiento de arboles multicast, mejorando la eficiencia del reenvío de paquetes bajo la familia de los mecanismos de distribución sin-estado basados en filtros de Bloom. Los resultados demuestran que D-MPSS (Depth-Wise Multi-Protocol Stateless Switching) es un método efectivo para deshacerse de las anomalías de reenvío, mientras se consigue una alta eficiencia de la transmisión, y que supera a MPSS.

Los autores que propusieron D-MPSS, llegaron a la conclusión que en MPSS ocurre lo siguiente:

- El conjunto de elementos a ser codificados podría ser demasiado grande (n = número total de enlaces del árbol).
- Esto implica la necesidad de grandes filtros de Bloom (BFs), para mantener la *fpr* baja.

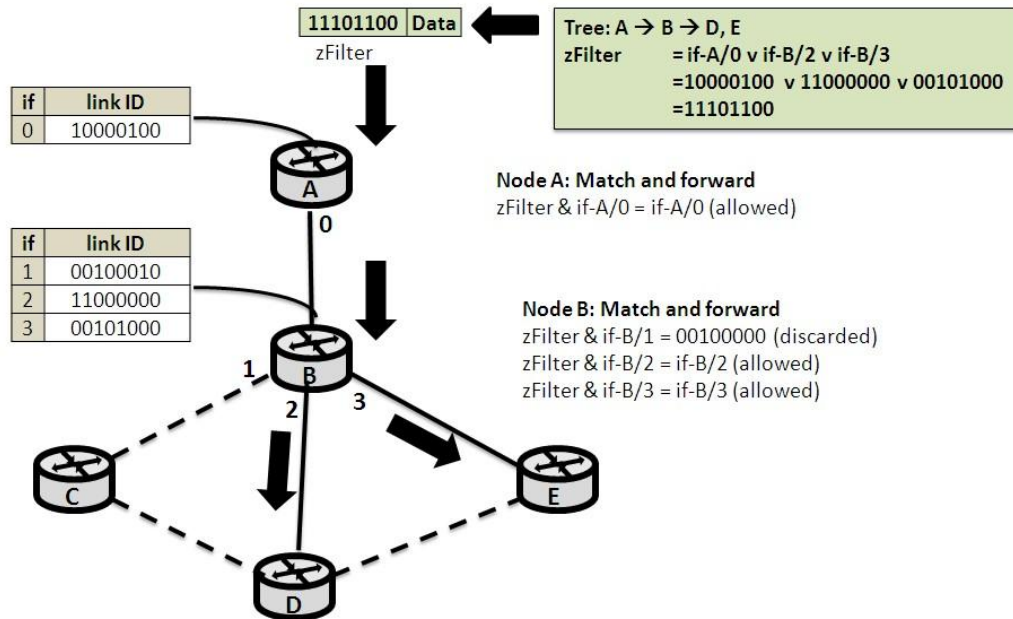


Figura 3.5.: Formación del zFilter y decisión de reenvío en MPSS

[GM12].

- Incluso los BFs relativamente grandes no pueden conseguir que *fpr* = 0. Los falsos positivos cercanos a la parte superior del árbol tienden a causar más severamente anomalías de reenvío.

Dado esto, se propone tener filtros de Bloom apilables, para reducir el espacio de elementos que se insertan en el BF. Para este fin se crean BFs individuales para cada nivel del árbol de enrutamiento. Así, un filtro de Bloom BF_i resume los enlaces pertenecientes al nivel de profundidad i -ésimo del árbol. Una vez que se computa el árbol de camino más corto, una pila de BFs es recolectada por cada hoja (Fig.3.6). Cada pila está compuesta de identificadores de enlace individuales desde la hoja hasta el nodo origen. Luego de tener todas las pilas juntas de cada una de las hojas, el nodo origen realiza una operación OR de los filtros individuales correspondientes a cada nivel de profundidad. Adicionalmente, se propone utilizar menos bits para codificar los enlaces de los niveles superiores, por lo cual se asumen nuevos BFs de longitud variable. Es decir si en principio los BF tenían una longitud de 64 bits estos podrían ser reducidos a la mitad, lo que permitiría una reducción de la cabecera del paquete.

Otro método que consiste en una variante de enrutamiento con filtros de Bloom, ha sido propuesto en [AR12], en el cual los autores proponen el esquema Counting Bloom Filter

3.1 Enrutamiento en Redes de Sensores Inalámbricos (WSN) / Capa de Red

Routing (CBFR). El esquema CBFR utiliza el protocolo CTP (Collection Tree Protocol) tanto para reenviar el paquete hacia el coordinador, como para ser reenviado entre los nodos intermedios. La idea central es que, cuando se tienen nodos móviles, se pueda eliminar del filtro de Bloom algún nodo que ya no está en

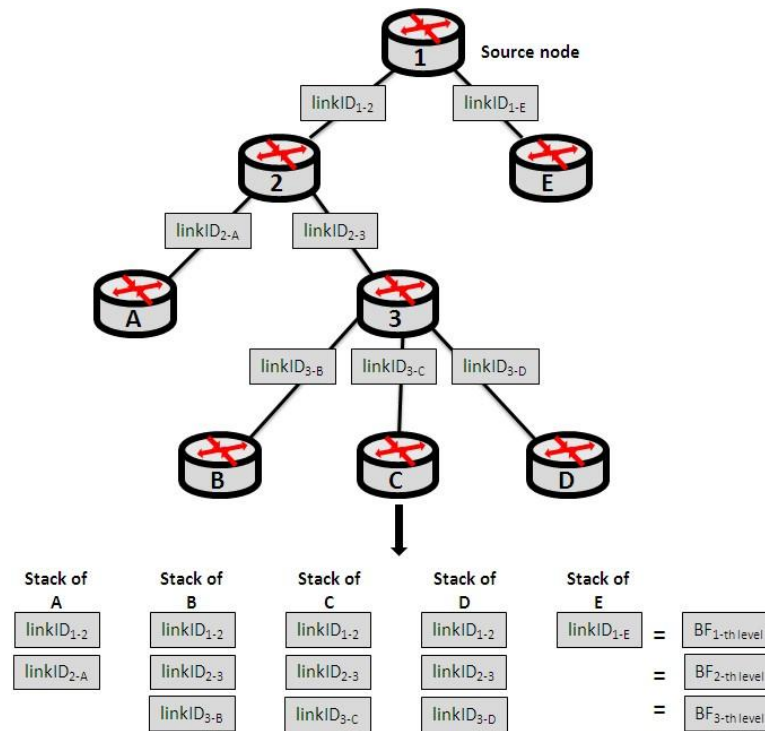


Figura 3.6.: Ejemplo de construcción de la pila de filtros de Bloom [GM12].

la ruta, a través del olvido gradual (que será explicado en el siguiente párrafo). En ámbitos generales este esquema facilita las siguientes operaciones: la inserción de nuevos elementos dentro del filtro y la eliminación de un elemento del filtro. Esto último no es posible en MPSS [AZN10] ni en D-MPSS [GM12].

En la inserción de elementos, un BF es típicamente implementado como un arreglo de m números enteros (a diferencia de las anteriores propuestas, en que el BF es un array de bits), los cuales son iniciados con valores de '0'. Al igual que en los otros esquemas, se van insertando los elementos en el array, las funciones hash h calculan el elemento insertado. El filtro de Bloom se crea en base a funciones hash aplicadas a la dirección del nodo (en este caso, un nodo móvil ZigBee).

En la Fig.3.7 se muestran dos procesos de elaboración de filtros de Bloom. El primero es el proceso de elaboración de un filtro de Bloom simple (BF), y el segundo el proceso de inserción y borrado de un filtro de Bloom contable. En la primera parte de la figura se insertan las direcciones id1, id2, id3 e id4 en un nuevo filtro de Bloom mediante una

operación OR. Cuando se realiza la consulta de si una dirección está incluida en este nuevo filtro generado, se realiza una operación AND; si el resultado es la misma dirección, significa que dicha dirección está incluida como un elemento del filtro. En este tipo de filtro, el borrado de una dirección es casi imposible, pues se estaría perdiendo posible información de otros elementos que habían sido insertados.

3.2 Conclusiones

En la parte derecha de la Fig.3.7 se hace uso de un filtro de Bloom modificado contable, conocido como Counting Bloom Filter (CBF). Este filtro en realidad ya no es del todo un filtro de Bloom, porque se trata de un arreglo de números enteros y no sólo un arreglo de bits. En este filtro modificado, los identificadores de los elementos a insertar siguen siendo arreglos de bits, pero la operación de inserción consiste en una operación de suma en vez de una operación OR. La ventaja principal de los CBFs es que en este tipo de filtros es posible el borrado de elementos. Para hacerlo, sólo es necesario restar la dirección a borrar. Esto es muy útil si en la red hay dispositivos que cierto tiempo están en la red y luego se desactivan o se reubican.

Filtros de Bloom(BFs)								Filtro de Bloom Contable(CBFs)							
Id1:	0	1	1	0	0	0	0	Id1:	0	1	1	0	0	0	0
Id2:	0	0	1	0	0	1	0	Id2:	0	0	1	0	0	1	0
Id3:	1	0	0	0	1	0	0	Id3:	1	0	0	0	1	0	0
Id4:	1	0	1	0	0	0	0	Id4:	1	0	1	0	0	0	0
Operación OR								Operación OR							
Filtro:	1	1	1	0	1	1	0	Filtro:	2	1	3	0	1	1	0
Filtro:	1	1	1	0	1	1	0	Filtro:	2	1	3	0	1	1	0
Consulta Id3:	1	0	0	0	1	0	0	Borrado Id2:	0	0	1	0	0	1	0
Verificación:	1	0	0	0	1	0	0	Operación AND/Resta							
Filtro:	1	1	1	0	1	1	0	Verificación:	2	1	2	0	1	0	0
Borrado Id2:	0	0	1	0	0	1	0								
Verificación:	1	1	1	0	1	1	0								

Figure 3.7.: Inserción y borrado de un elemento[Fuente: Elaboración Propia].

En la propuesta realizada por los autores, se ve que es posible el borrado de elementos en un filtro, sin embargo nace la conjetura sobre ¿qué pasaría si se tiene un falso positivo? Si se llega a realizar las verificaciones y se comprueba que el elemento está dentro del filtro, el sistema lo admitirá y trabajara con él. Sin embargo, si borramos dicho elemento del filtro, y éste resulta que en realidad no es un elemento del grupo (un falso positivo), lo más probable es que se pierda información importante, deteriorando el filtro acumulado debido a un elemento que no pertenece a la red. En cambio, al usar filtros de Bloom normales, se puede llegar a interactuar con los falsos positivos, más el borrado no es admitido, ya que al realizarlo borrado, se pueden perder los elementos ingresados.

3.2. Conclusiones

En la mayoría de trabajos expuestos, se han realizado distintos tipos de estudios, ya sea sobre el estándar IEEE 802.15.4/ZigBee, como sobre redes de sensores inalámbricos (WSN).

Es importante ver que en algunos trabajos se propone el uso de filtros de Bloom o la reducción de las tablas de enrutamiento, con la finalidad de mejorar las redes pre-establecida.

Capítulo 3

Vemos que cada investigador propone nuevos métodos de cómo mejorar el desempeño del protocolo ZigBee. En algunos trabajos hacen uso de tablas de enrutamiento que logran abarcar lo esencial que caracteriza a las redes inalámbricas; gran parte de investigadores encuentran soluciones distintas para mejorar el protocolo de enrutamiento, lo cual permite mejorar el rendimiento y aumentar el tiempo de vida de la red. Sin embargo, no aportan mucho en a la disminución de la sobrecarga de la red. A través del estudio realizado se ha considerado tomar algunos métodos que han aplicado algunos autores para poder elaborar nuestro trabajo.

La mayoría de las redes se caracterizan por su capacidad de reorganizarse, en caso hayan perdido su conexión hacia una ruta antes establecida. El proceso de reorganización a su vez genera un gran costo, como tráfico y utilización de recursos (ya sea la memoria, la baja capacidad de transferencia de información, entre otras cosas). Por tal razón, muchos investigadores concentran sus investigaciones en cómo mejorar el desempeño de la red, pero a través de optimizaciones de enrutamiento y distribución de tablas. En cierto grado arreglan el problema fundamental, mas no muestran una mejora al momento de enviar paquetes a todos sus nodos. Mientras siguen entablando comunicaciones broadcast, sólo lograrán que nuestra red siempre ocupe recursos innecesarios.

La literatura revisada gira en torno a dos temas: el uso de filtros y las comunicaciones multicast. Respecto al uso de los filtros de Bloom, otorgan un beneficio muy importante, ya que permite usar los recursos limitados más eficientemente, mejorando el rendimiento de los nodos de la red. Al reducirse las tablas de enrutamiento, los nodos procesan más rápidamente los paquetes, además ya no tienen una sobrecarga en el uso de memoria. Indirectamente, se podría incluso conseguir un menor uso de energía de alimentación para estos nodos (componente escaso en redes de sensores).

4. Propuesta de Solución

4.1. Propuesta de mecanismo de enrutamiento usando Filtros de Bloom

En la propuesta de solución para un nuevo esquema de enrutamiento mediante filtros de Bloom, se ha involucrado aspectos nuevos no antes considerados en ZigBee. Todos estos aspectos que a continuación se exponen han permitido conseguir un mejor funcionamiento para esta tecnología emergente. Se han estudiado diversas posibles soluciones para la

mejora. Un aspecto crítico a mejorar son las tablas de enrutamiento generadas en las redes ZigBee, las cuales pueden llegar a ser muy grandes, sobre todo cuando la red tiende a realizar actualizaciones cada cierto periodo.

La agregación de nuevas rutas, la agregación o desagregación de nodos o equipos, incrementan el tamaño de las tablas de enrutamiento en ZigBee. La principal carga de la red se focaliza en el nodo padre y sus nodos contiguos, los cuales muchas veces hacen uso excesivo de los recursos limitados.

En las redes de prueba hicimos uso de filtros de Bloom, puestos en la cabecera de los paquetes. El primer paso que se realizó fue la verificación del radio de cobertura de cada nodo. En este ejemplo se muestra una misma potencia de radiación en cada nodo; este proceso de alcance permite detectar si dentro del radio de cobertura, el nodo puede detectar otros dispositivos. Para conocer que un nodo se puede conectar con otro dispositivo dentro de su radio de cobertura, se recurrió al uso de la fórmula matemática mostrada a continuación, que permite conocer si un punto está dentro del área de una circunferencia.

$$(x - h)^2 + (y - k)^2 = r^2 \quad (4.1)$$

En la Figura 4.1 tenemos nodos que transmiten datos con la misma potencia de radiación, en esta imagen se verifica de forma visual lo antes argumentado. La gráfica muestra nodos que transmiten dentro de un radio de cobertura de forma individual, y la captura de nodos dentro de su radio de cobertura.

El desarrollo de la solución propuesta la guiaremos a través de un ejemplo. En una red inalámbrica, es importante conocer si algún nodo o dispositivo detecta a otro dentro su rango de cobertura. En la Figura 4.2 se consideró un mismo rango de cobertura para todos los nodos. El nodo R1 logra detectar otros nodos dentro de su

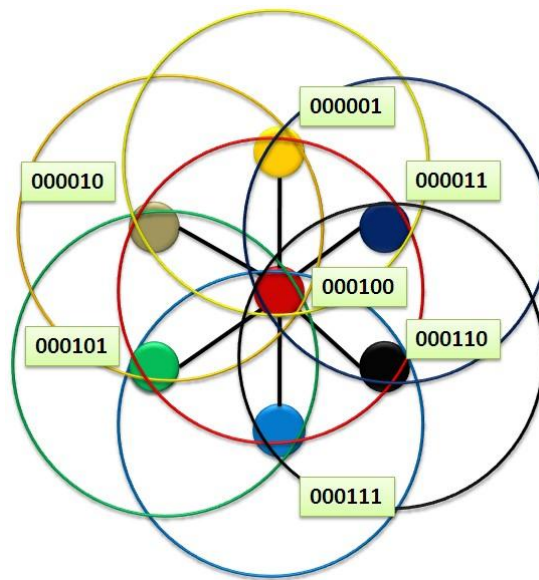


Figura 4.1.: Cobertura individual según la potencia de transmisión [Fuente: Elaboración Propia] .

radio de cobertura. La radiación del nodo *R1* está representada por un círculo más grande que captura y/o detecta otros nodos cerca al nodo *R1*.

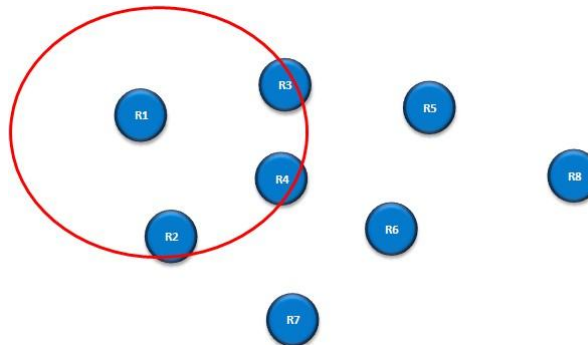


Figura 4.2.: Rango de cobertura del nodo *R1* [Fuente: Elaboración Propia].

Al igual como hizo el reconocimiento previo el nodo *R1* de posibles nodos detectados, cada nodo contenido en la red hace el mismo procedimiento para detectar a otros nodos. Luego del reconocimiento previo de nodos, cada nodo genera un ID en forma de filtro de Bloom, de forma aleatoria. El ID consiste en un arreglo de bits, la cantidad de 0's es mucho más grande que la cantidad de 1's.

Luego de la identificación de cada nodo con su nuevo ID, estos hacen un reconocimiento previo de los nodos vecinos que están a su alcance en su radio de cobertura, como se muestra en la Figura 4.3.

4.2 Formación de árboles multicast

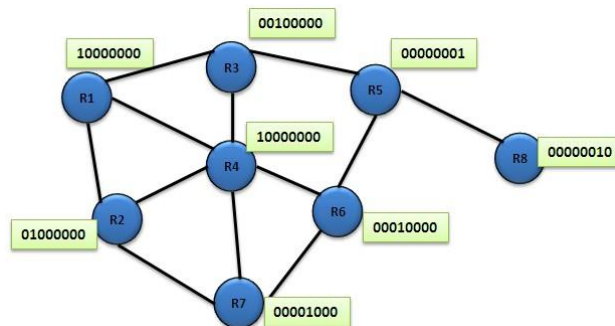


Figura 4.3.: Asignación aleatoria de un ID a cada nodo y reconocimiento de nodos vecinos [Fuente: Elaboración Propia].

4.2. Formación de árboles multicast

El estándar IEEE 802.15.4/ZigBee posibilita la comunicación punto a punto (*Unicast*), y/o punto a multipunto (*Broadcast*). El estándar no especifica la existencia de una comunicación de un nodo a un grupo específico de nodos (*Multicast*). Sin embargo este tipo de comunicación es necesaria para una red de sensores.

En las redes actuales es muy importante conocer la posibilidad de entablar comunicación con todos sus dispositivos de las distintas maneras existentes, la finalidad muchas veces es limitar el uso innecesario en las redes. Es decir, si se requiere enviar un mensaje a un grupo específico de nodos, este debería de llegar solo a esos nodos, esto hace que sólo use recursos necesarios para llegar sólo a esos nodos en específico.

En el caso que se requiera enviar datos a un grupo específico de dispositivos y/o nodos y no se tiene implementada una función o manera adecuada, este mensaje viajaría por toda la red haciendo uso innecesario del ancho de banda, reduciendo la capacidad de la energía de las baterías, entre otros aspectos, por lo cual es importante definir formas adecuadas de entablar comunicaciones con los equipos que solo se requieren que participen.

En las redes inalámbricas actuales, un dispositivo puede funcionar de distintas maneras, como también cada dispositivo tiene la potestad de seleccionar y organizar su propia red. En las redes ZigBee se tiene una estructura ya antes formada, en la que el nodo coordinador recibe las tablas de enrutamiento de sus nodos hijos, este

determina sólo una única red árbol, y a través de éste realizan todas las comunicaciones a los demás nodos.

En esta sección hablamos sobre formación de árboles multicast, considerando los conceptos básicos de ZigBee. Hemos considerado la posibilidad de que cada nodo tenga la misma oportunidad de trabajar y funcionar distintamente en distintos entornos. Es decir, que todos los nodos de la red puedan funcionar como nodo padre, o como un nodo hijo o nodo intermedio, de acuerdo a la petición multicast.

Para probar la solución propuesta, se han generado grupos aleatorios multicast, donde cada grupo tiene la posibilidad de generar su propio árbol. La generación de este tipo de grupos permite entablar comunicación sólo con los dispositivos que pertenecen al grupo, y esto permite que no se desperdicien recursos al momento de hacerlo, como el ancho de banda, y otros recursos limitados que caracterizan a ZigBee.

Ahora que conocemos un poco más sobre la formación de los árboles multicast, a continuación mostramos cómo es el proceso de formación de grupos multicast. En nuestro ejemplo seleccionamos al nodo *R2* como nodo padre, y *R1*, *R4*, *R3* y *R5* nodos pertenecientes al grupo multicast almacenados en el grupo 1 (grup#1). Luego se toman sus ID'S o filtros de Bloom de manera independiente y se aplica una operación "OR".

La Figura 4.4 muestra la operación OR de los filtros seleccionados para la formación de un grupo multicast. Todos los ID's de los nodos seleccionados son agrupados en un nuevo filtro de Bloom. En la Figura 4.4 se muestra el resultado de operar los nodos participantes en un nuevo filtro.

Grup. #1	R1	1	0	0	0	0	0	0	0
	R4	1	0	0	0	0	0	0	0
	R3	0	0	1	0	0	0	0	0
	R5	0	0	0	0	0	0	0	1
Operación OR		+	+	+	+	+	+	+	+
Grup. #1	LabelBF	1	0	1	0	0	0	0	1

Figura 4.4.: Formación de un grupo multicast [Fuente: Elaboración Propia].

Luego de encontrar el nuevo filtro de Bloom perteneciente al primer grupo multicast, el nodo origen de la petición multicast selecciona la mejor ruta para alcanzar a los nodos del grupo, creando un árbol. Para esto, se utiliza el método post-orden. Una vez conformado todo el árbol, se generan nuevos filtros de Bloom en cada nodo, resumiendo los nodos hijos y hojas que están debajo de cada nodo intermedio (con operaciones OR) , de acuerdo a la jerarquía en que se encuentren.

En la Figura 4.5 se observa que el nodo padre está ubicado en la parte más alta del árbol, conocido como el nodo coordinador. Los nodos intermedios son conocidos como nodos hijos. En ZigBee, los nodos intermedios con los enrutadores, debido que es el camino intermedio que permite conectar mas dispositivos. Los nodos que se encuentran en la

parte más inferior son los nodos hojas, que en la práctica son usualmente los dispositivos sensores, que capturan la información y la envía al nodo coordinador.

El funcionamiento de los filtros resúmenes almacenados en los nodos hijo (denominados Children Bloom Filter, CBF) consiste en que, al momento de recibir un paquete, se hagan operaciones AND entre el filtro de Bloom del paquete y el CBF.

4.2 Formación de árboles multicast

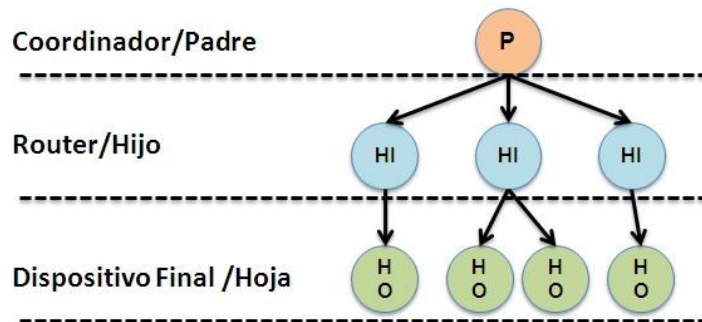


Figura 4.5.: Tipos de nodos en ZigBee [Fuente: Elaboración Propia].

En la Figura 4.6 se muestra un ejemplo de una red, las líneas resaltadas con color rojo indican la ruta (rutas virtuales) seleccionada para entablar una comunicación multicast, también en la figura se muestra el resumen de los nodos hijos participantes de la red.

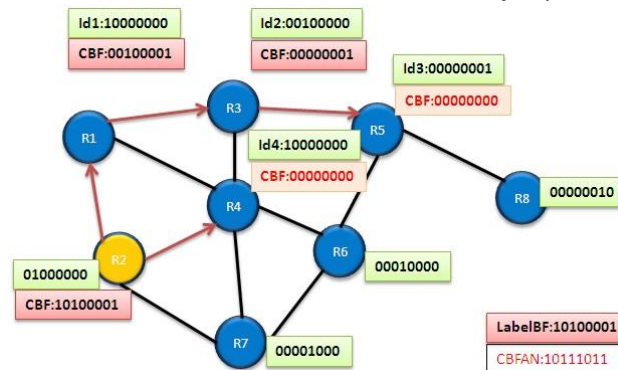


Figura 4.6.: Filtros resúmenes de nodos participantes [Fuente: Elaboración Propia].

Para la formación de los filtros resúmenes de los distintos nodos, se utiliza el método post-orden, el cual permite recorrer un árbol desde los nodos hoja hasta el nodo padre. En la Figura 4.6 se denota con líneas rojas el árbol del grupo multicast. Los nodos participantes en el grupo multicast son R2, R1, R3, R4, y R5. En este ejemplo R2 funge de nodo coordinador, debido a que es el origen de la petición multicast; R5 y R4 son los

nodos hoja. Cada nodo tiene un filtro de Bloom que hace las veces de su identificador en la red (resaltado de color verde).

Para la construcción de filtros resúmenes (CBF) es importante conocer la ubicación de cada nodo dentro del árbol. *R5* es un nodo hoja y es hijo del nodo *R3*, por lo tanto su filtro CBF contiene un valor igual a 00000000, ya que no tiene nodos hijos. En cambio el nodo *R3*, tiene como nodo hijo a *R5* y como nodo padre a *R1*, por lo que el nodo *R3* almacenará el filtro resumen del nodo *R5*, el cual es 00000001.

Luego el nodo *R1* tiene como nodo hijo a *R3* el cual es padre de *R5*, por lo tanto *R1* almacena un nuevo filtro producto de hacer una operación OR de los filtros de *R3* y *R5* = 00100001.

El nodo *R4* es un nodo hoja, entonces su filtro es 00000000. El nodo padre de *R4* es *R2*, el cual es el nodo coordinador del grupo multicast, por lo cual tiene contenido el resumen de todos los nodos involucrados de su red como *R1*, *R3*, *R4*, y *R5*, estos son sus nodos hijos, y mediante la operación OR se obtuvo el filtro resumen 10100001.

En la siguiente sección se muestra el tipo de contenido portado en el paquete que está incluido en la cabecera al momento de entablar una comunicación multicast. También se muestra cómo el paquete es evaluado de acuerdo a los resúmenes CBF. En la sección que acabamos de concluir vimos que además de contener un filtro de Bloom que contiene la operación OR del grupo multicast, se incluyó el número de grupo, el cual hace viable las comunicaciones dentro del propio grupo multicast. Por ejemplo, si el paquete multicast tiene el #1, el paquete tiene la capacidad de interactuar sólo con resúmenes de nodos hijos pertenecientes al primer grupo #1 de lo contrario el contenido será descartado.

4.3. Envío del paquete multicast

La cabecera del paquete multicast contiene el número de grupo y el resumen de los nodos participantes representado en un filtro de Bloom. En la Figura 4.7 se muestra la forma del paquete multicast.

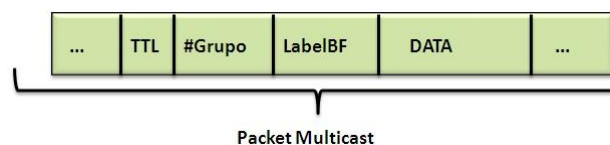


Figura 4.7.: Paquete multicast [Fuente: Elaboración Propia]

Recapitulando, la idea central es cómo tener una comunicación multicast entre nodos inalámbricos. Los nodos participantes son miembros de grupos multicast, cada grupo tiene un nodo coordinador, desde el cual se traza todo el árbol para dicho grupo. Se

cuenta con filtros resúmenes de los nodos intermedios e hijos, el cual facilita la conmutación salto a salto dentro de la red. Una de las finalidades es reducir el tiempo de búsqueda al momento de interactuar con todos los nodos pertenecientes a un grupo multicast. También el objetivo es reducir el número de búsquedas que se hacen en las tablas de enrutamiento de los nodos.

Cada vez que el paquete visite un nodo se verificará si éste es parte del árbol multicast. En cada salto se registra en cada nodo el paso de dicho paquete, de tal modo que si se vuelve a recibir el mismo se rechace. Si resulta bien la verificación, el paquete se seguirá propagando. Se restringe el número de saltos que un paquete puede

4.3 Envío del paquete multicast

dar, con un contador TTL que se actualiza salto a salto. Estas medidas ayudan a evitar que nuestra red colapse por el abuso de recursos innecesarios formados por bucles y duplicidad de paquetes.

La verificación de la cabecera del paquete con forma de filtro de Bloom se hace mediante una operación *AND* entre el filtro de Bloom y el resumen del nodo (CBF). Si esta operación da como resultado el mismo resumen del nodo, entonces el paquete es admitido y se sigue difundiendo al siguiente salto, siempre y cuando se verifique que el mismo pertenece al grupo multicast.

En la Figura 4.8 se grafica cómo el paquete realiza las verificaciones antes de seguir difundándose. En la parte superior izquierda podemos ver el paquete multicast, con la indicación de un grupo específico y su filtro de Bloom. Primero se visita el primer nodo (*R2*) o nodo padre, dicho nodo verifica que el paquete es del grupo #1, por tal razón se revisará las entradas en la tabla de dicho grupo, es decir el resumen de sus nodos hijos. En la imagen se resalta la tabla de enrutamiento del nodo *R2*, en el rectángulo rojo la selección del grupo, además del filtro resumen.

Para la verificación del reenvío del paquete, se toma la etiqueta LabelBF y se aplica junto con el filtro resumen de *R2* la operación *AND*. Dado que el resultado de dicha operación nos devuelve el mismo filtro resumen de nodos hijos contenidos en este grupo (como lo mostrado en la figura), se procede a entregar el paquete y se sigue difundiendo el mismo.

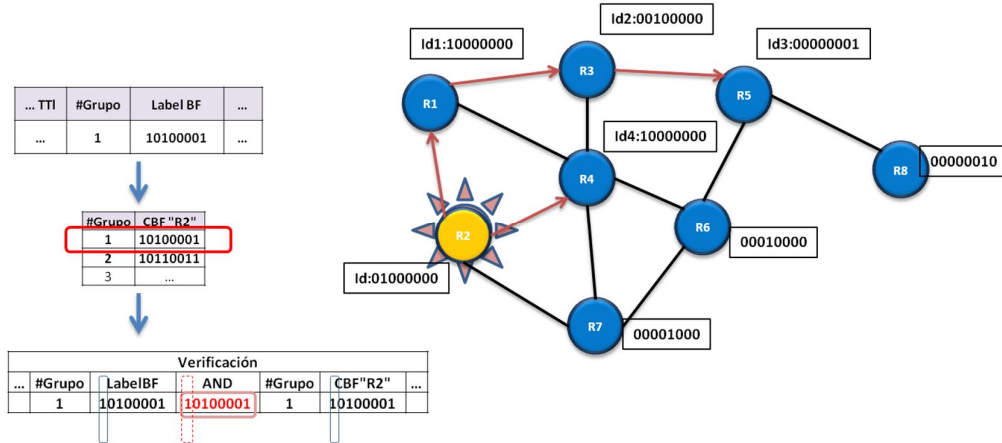


Figura 4.8.: Verificación de la cabecera del paquete en el nodo R2 [Fuente: Elaboración Propia].

En la Figura 4.9, se muestra el mismo proceso de verificación con el siguiente nodo (R1). Si los resultados de la verificación no dieran como resultado el filtro resumen de los nodos hijos, el paquete se rechazaría anulándose la posibilidad de continuar difundiendo. Otro parámetro que determina la duración del paquete es el tiempo de vida del mismo (dado en el máximo número de saltos que el paquete puede dar), que es tomado acorde a la profundidad del árbol multicast.

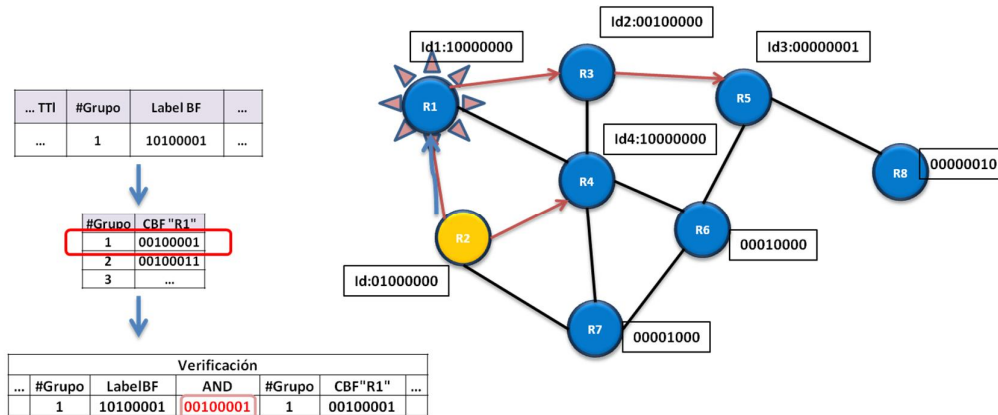


Figura 4.9.: Verificación de la cabecer del paquete en el nodo R1 [Fuente: Elaboración Propia].

Podemos ver claramente que, cuando se somete a una actualización a un grupo definido de nodos, no es necesario el uso de toda la red, lo cual ayuda a moderar el uso del ancho de banda y alargar la vida de la batería de otros dispositivos que no participaron. Esta propuesta logra optimizar el ancho de banda de la comunicación entre los dispositivos

ZigBee, y a la vez permite optimizar el uso de algunos recursos limitados, dando mayor estabilidad y seguridad en torno a las aplicaciones que se vayan a implementar (dado que en este sistema los grupos pueden aislarse en su funcionamiento, de acuerdo a su respectivo número de grupo, sin generar comunicaciones broadcast a todos los nodos).

5. Simulaciones, resultados y comparaciones

5.1. Software

El software utilizado para implementar la propuesta de solución, está desarrollado sobre el lenguaje de programación Java. Se ha seleccionado este lenguaje de programación debido a la familiaridad y frecuencia de uso, consecuentemente da oportunidad que otros investigadores puedan reutilizar el código desarrollado.

Para lograr simular la solución propuesta, hemos hecho uso de:

- Java (JSDK, JRE)
- Eclipse

5.1.1. Eclipse

Eclipse es una plataforma de programación para el lenguaje Java[Ecl13], compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados, como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse. Eclipse tenía en un inicio la Common Public License, pero después fue relicenciado bajo la Eclipse Public License. Ambas son licencias de software libre, pero son incompatibles con la licencia pública general de GNU (GNU GPL).

5.1.2. Java

Es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Permite que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que

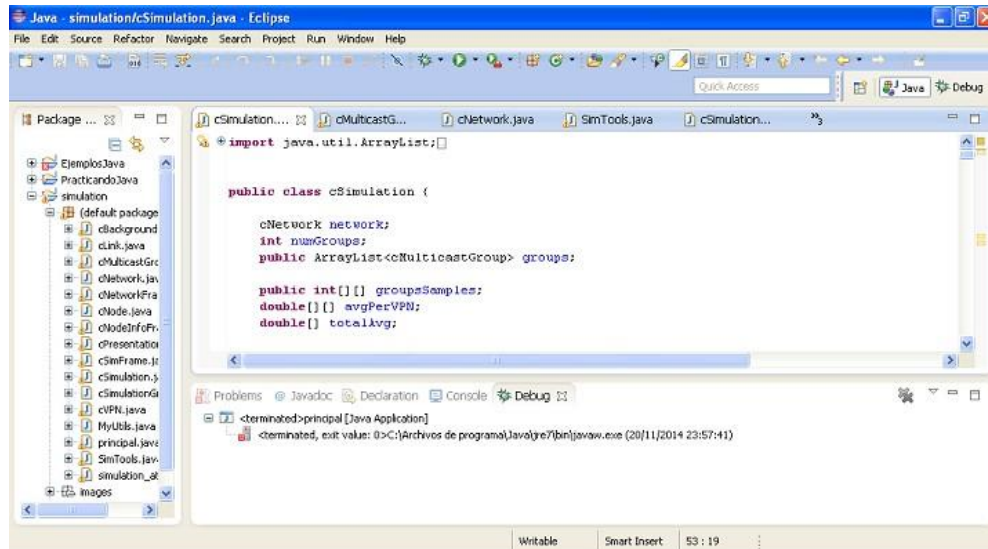


Figura 5.1.: Eclipse [Ecl13].

ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados hasta la fecha.

La compañía Sun desarrolló la implementación de referencia original para los compiladores de Java, máquinas virtuales, y librerías de clases en 1991 y las publicó en 1995. A partir de mayo de 2007, en cumplimiento con las especificaciones del Proceso de la Comunidad Java, Sun volvió a licenciar la mayoría de sus tecnologías de Java bajo la Licencia Pública General de GNU. Otros también han desarrollado implementaciones alternas a estas tecnologías de Sun, tales como el Compilador de Java de GNU y el GNU Classpath[Mic13b].

5.2. Simulación implementada

En el desarrollo de la plataforma de simulación, ha sido necesario establecer diferentes tipos de redes, las cuales fueron sometidas a distintas pruebas. Además, estas redes, como se muestra en el *Anexo1* en la clase *cNetwork*, tienen una topología predeterminada. Esto se diseñó de esta manera, para poder establecer comparaciones y calcular las medias de la eficiencia de la propuesta, dado que, si las redes no tuvieran una topología fija sino cambiante, no se podrían interpretar los resultados.

Para calcular los árboles multicast, se utilizó el algoritmo de Dijkstra, el cual permite encontrar la ruta más corta desde un nodo origen a otros nodos. Dicho método se

encuentra en el *Anexo1* en la clase *cNetwork*. Para el establecimiento del árbol multicast fue necesario usar el método *postorden*, que permite realizar un recorrido

5.2 Simulación implementada

de toda el árbol y determinar que nodos o dispositivos son hojas y quienes son nodos padres.

Respecto a la cobertura de radio de los nodos inalámbricos, en el *Anexo1* en la clase *SimTools*, se ha definido el método matemático que permite saber si un punto esta contenido dentro de un círculo. En el protocolo ZigBee se considera que un nodo es alcanzable si está dentro del radio de cobertura de otro nodo.

5.2.1. Generación de ID's de forma aleatoria

Se han construido varias clases para la inicialización de la red, que incluyen procesos de generación de los nuevos ID's para cada nodo (en la clase *SimTools*), los cuales son generados desde cada dispositivo, por lo cual no tienen que esperar a un nodo superior o coordinador para que realiza la petición de asignación de direcciones. Los ID's aleatorios generados por cada nodo son filtros de Bloom, donde "*m*" determina el tamaño del arreglo de bits y "*k*" el número de funciones hash y/o elementos que tendrá el direccionamiento dentro del filtro (en la práctica, el número de bits puestos en 1). Consideramos que el tamaño máximo de un filtro de Bloom es de $m = 128$, debido a que es un número cercano al tamaño de cabecera de ZigBee, y la cantidad de elementos insertados en estos filtros varía desde un $k = 1$ hasta $k = 5$. Para generar un nodo ID único, de forma aleatoria: *m*: Número de bits del ID *k*: Número de 1's (tienen que ser

- muy pocos)

$$\text{Numero de IDs distintos'} = \frac{m!}{[k!(m-k)!]}$$

Por ejemplo:

Si $m = 128$ y $k = 3$, se obtienen hasta 341,376 distintos ID's. Evidentemente, a pesar de que la probabilidad de generar aleatoriamente dos ID's iguales es muy baja, sí podría ocurrir esta incidencia (también con una muy baja probabilidad).

Una de las mejoras de esta propuesta es la asignación de direcciones, de forma menos compleja que en ZigBee. En la propuesta de solución, cada nodo tiene la capacidad de generar su propio direccionamiento de forma aleatoria. Luego, es necesario que el nodo fuente calcule el árbol, mediante Dijkstra. Para efectos de la simulación, se usa el método *postorden* para recorrer el árbol calculado y determinar los padres y nodos

hijos. Dicho método se encuentra implementado en la clase *cNetwork* y *cMulticastGroup*.

También para efectos de simulación, se utiliza el método *postorden* para recolectar los filtros de los nodos hijos y resumirlos en los CBF en cada nodo. Esto se encuentra codificado en la clase *MyUtils*.

5.2.2. Descentralización de las transmisiones multicast

Hasta el momento se tienen implementados tres puntos de los cuatro propuestos en el planteamiento del problema. El último está referido a la centralización de toda comunicación sobre un nodo coordinador. Recordemos que en ZigBee toda comunicación viaja sobre una misma topología árbol, y si dos nodos contiguos quieren establecer comunicación, esta debe viajar primero al coordinador, el cual reenvía los paquetes entre los nodos. Además, las redes ZigBee solo pueden entablar dos tipos de comunicación, una de tipo punto a punto (*Unicast*) y otra de punto a multipunto (*Broadcast*).

En las simulaciones se generan grupos multicast, de un nodo a un cierto grupo de nodos. Se tiene una manera fija y una manera variable de generar dichos grupos. En la distribución fija, se predefine la cantidad de dispositivos que pueden conformar el grupo multicast, mientras que en una distribución variable el número de miembros está definido por una probabilidad uniformemente variable. Los nodos se escogen aleatoriamente.

En la clase *cSimulation* se han implementado los tres métodos para generar grupos multicast. Un dispositivo puede ser un nodo hijo o coordinador, cada vez que se tiene una petición de grupo multicast, el dispositivo asignado como coordinador es aquel que tiene que enviar datos a los demás nodos, a través de la ruta-árbol que calcula. En consecuencia un mismo dispositivo puede tener funcionamientos distintos en diferentes escenarios, por lo cual, los nodos involucrados en una red, tienen la posibilidad que en cierto momento cumplan el rol de nodo coordinador, manteniendo sus direccionamientos generados al momento de establecerse la red árbol.

5.2.3. Simulación de las transmisiones de paquetes en los grupos multicast

Para la transmisión de los paquetes en los grupos multicast, en cada salto se evalúa si el número de grupo almacenado en la cabecera del paquete, está en la tabla de enrutamiento del nodo. Esto ayuda a que el paquete sólo llegue a los miembros del grupo.

Se realiza la simulación del envío de paquetes dentro de un grupo multicast (ver *Anexo1* en la clase *cMulticastGroup*), para lo que se ha codificado el algoritmo de la Algoritmo 5.1, para la verificación de los paquetes en cada nodo. Cuando se envía el paquete, en cada nodo se compara el filtro de Bloom de la cabecera con el filtro resumen del nodo actual, mediante la operación “AND”. Se debe obtener entonces un filtro resumen igual al del nodo actual, de serlo, el paquete es aceptado por el nodo y reenviado, de lo contrario se descarta. Si el nodo contiene un resumen en forma de arreglo de ceros, el resultado es nulo, y se deja de difundir el paquete, pues eso quiere decir que se trata de un nodo hoja.

5.2 Simulación implementada

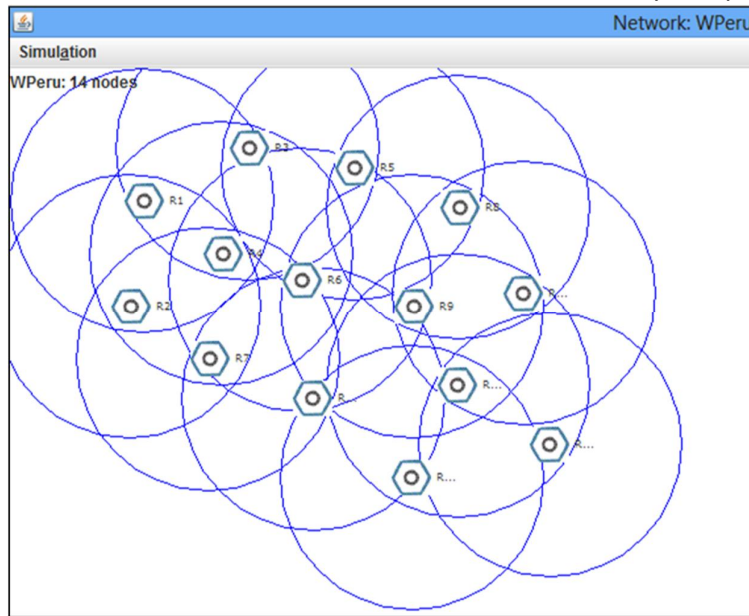


Figura 5.2.: Radiación de cada nodo [Fuente: Elaboración Propia].

5.3. Ejecución de las simulaciones

Mediante el uso de una computadora y la previa instalación del software, se procedió a implementar el código. Luego se procedió a ejecutar dicha implementación, para verificar que el funcionamiento sea óptimo, consecuentemente se realizaron pequeños ajustes, los cuales lograron probar la solución propuesta. Además, el software usado permite desarrollar aplicaciones o entornos gráficos para visualizar la red en la cual estamos interactuando, en la Figura 5.2 logramos ver la red *WPeru'*, la cual muestra el mismo radio de cobertura en cada uno de sus nodos, además se logra ver como su potencia de radiación llega a contactar otros nodos, para tal efecto el código que permite esta representación grafica, también está ubicada en el *Anexo1*.

En el desarrollo de la simulación también se incluyó la generación de tablas de nodos vecinos, que permiten corroborar con parte de la gráfica mostrada en la imagen anterior. A continuación se muestra la Tabla 5.1 haciendo referencia a los resultados de los nodos vecinos para una red previamente establecida y/o seleccionada, los resultados de la simulación son como lo muestra en la Figura 5.3.

Los resultados generados por la simulación de posibles nodos vecinos de la red propuesta, son mostrados en la Figura 5.3.

En la simulación implementada para redes Zigbee a través de filtros de Bloom, también se ha incluido resultados, como nodos predecesores, los cuales permiten bosquejar la

red luego de que se evaluó con el algoritmo Dijkstra. En el *Anexo1* de este trabajo se incluye la parte de la programación que permitió obtener estos

5.3 Ejecución de las simulaciones

Node	1	2	3	4	5	6	7	8
1		x	x	x				
2	x			x			x	
3	x			x	x			
4	x	x	x			x	x	
5			x			x		x
6				x	x		x	
7		x		x		x		
8					x			

Cuadro 5.1.: Nodos vecinos de la red propuesta en la Figura 5.2 [Fuente: Elaboración Propia].

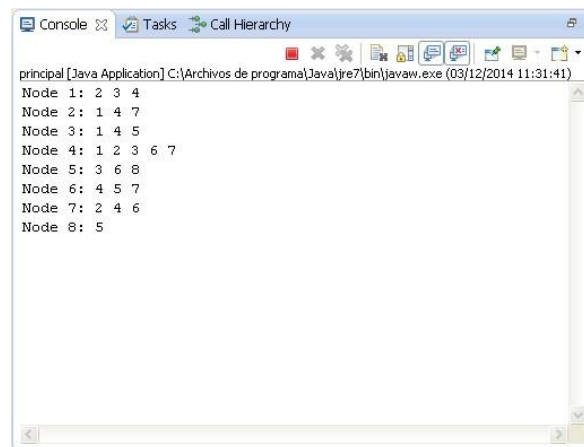


Figura 5.3.: Resultados de nodos vecinos de la red propuesta [Fuente: Elaboración Propia].

resultados, como también los resultados generados por el método post-orden, que son bosquejados en la Tabla 5.2 con sus filtros resúmenes. La tabla mostrada a continuación

muestra los IDs en cada nodo, también muestra filtros resúmenes de nodos hijos, es importante mencionar que cuando encontramos el valor de “0” en el filtro resumen, dan referencia que se está ante un nodo hoja, por lo cual no cuenta con nodos hijos. Sin embargo, cuando existe en el filtro más de un valor de “1”, referencian la existencia de nodos hijos.

También se hizo el uso de filtros resúmenes para las comunicaciones multicast, en el proceso de la generación de grupos multicast, se formaron grupos de manera estática como se muestra en el *Anexo1* (la clase *cSimulation*), pero se encontró métodos que nos permitirían generar grupos multicast a partir de la selección de nodos de manera aleatoria.

Para el proceso de generación de grupos multicast, el tipo de alcance propuesto es

Nodo	BF-ID	BF-Resumen
R1	10010000	00000000
R2	01000000	11111011
R3	00100000	00000000
R4	10000000	11111010
R5	01000000	00000000
R6	00010000	11011010
R7	00001000	01110001
R8	00000010	00000000
R9	10000010	01011010
R10	01100000	01010001
R11	01001000	00000000
R12	00011000	01001000
R13	01010000	00000000
R14	01001000	00000000

Cuadro 5.2.: Filtros resúmenes de nodos hijos [Fuente: Elaboración Propia].

de tipo variable. En la distribución de grupos multicast fija está basada en la generación de grupos de nodos tomados aleatoriamente de la red, en esta generación de grupos multicast uno determina manualmente la cantidad de nodos que quisiera que conforme el grupo. En una distribución variable, se logra obtener grupos multicast con distinta cantidad de nodos que la conforman, en una distribución variable de tipo uniforme la

probabilidad de que se considere un nodo en un grupo es la misma probabilidad que existe al considerar una red de nodos en un mismo grupo.

Las simulaciones no solo están basadas en generar filtros de Bloom para cada nodo, a lo largo de la simulaciones se ha involucrado la forma de trabajar con los mismos. En la generación de grupos multicast se tomaron los filtros independientes de cada nodo para generar uno nuevo, por consiguiente se realizó una prueba de envío de paquete a un grupo específico de nodos para verificar la factibilidad de funcionamiento del mismo. Además, se ve cuando se forma el grupo multicast, también se encuentra un nodo asignado, el cual arma su propia topología árbol, para cuando se envía un paquete multicast.

En la propuesta de solución se hace la mención de una cabecera, la cual está transportada dentro de un paquete. En la cabecera se tiene también un número entero, para representas el número de grupo multicast. Luego la cabecera cuenta con otro espacio donde se integra un filtro de Bloom (*etiquetalabelBF*), el cual ha sido generado por la selección aleatoria de nodos para formar un grupo multicast. Cada vez que se vaya a establecer una comunicación, este paquete visitará los distintos nodos de la red, la razón es para conocer si es el ruta adecuada, y además si el nodo está incluido en su paquete.

5.4 Simulaciones

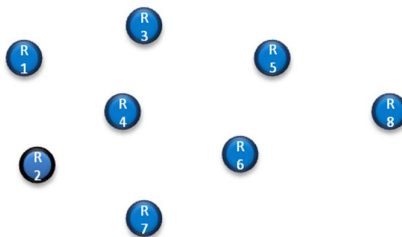


Figura 5.4.: Ubicación aleatoria de nodos, red *WArequipa*[Fuente: Elaboración Propia].

5.4. Simulaciones

Los resultados se han obtenido en base a las pruebas realizadas sobre distintas redes simuladas, sobre las cuales se han generado aleatoriamente 100 grupos multicast. La generación de estos grupos se realizó usando una distribución aleatoria uniforme de los nodos destino de los grupos. Se ha probado con distintos tamaños de direcciones de los nodos, las cuales son filtros de Bloom que varían en tamaño de $m = 8$ a $m = 128$. También se ha variado la cantidad de elementos insertados en los filtros (es decir, el número de 1's en el filtro), entre $k = 1$ y $k = 3$. Es importante recalcar que en una red ZigBee el tamaño máximo de nodos es de 255.

La finalidad es probar el rendimiento de nuestra propuesta en distintos escenarios de grupos multicast. Cuando se generan grupos multicast de distribución uniformemente variable, hay la probabilidad de que un grupo multicast esté formado sólo por dos nodos (en cuyo caso la comunicación entre origen y destino es prácticamente una comunicación unicast), muchos nodos o por todos los nodos de la red (broadcast).

Representación de la topología de red

La Figura 5.4 muestra un conjunto de nodos ubicados aleatoriamente en el espacio. Esta red es un conjunto de dispositivos que se comunican de manera inalámbrica, no existe un medio cableado que forma enlaces entre los dispositivos. En la Figura 5.4, se muestra la red que hemos denominado *WArequipa*, la cual esta *Anexa* en *cNetwork*.

La Figura 5.5 muestra el radio de cobertura inalámbrica de cada nodo de la red *WArequipa*, además se ven los posibles nodos que se encuentran dentro del rango de cobertura. Esta detección la realiza cada nodo de manera inalámbrica, luego, cada nodo actualiza su tabla de nodos vecinos. Para efectos de las simulaciones, los nodos vecinos son representados entre sí por enlaces (Figura 5.6), siempre y cuando entre dos nodos se de la siguiente condición:

$$q \quad (x_c - x_p)^2 + (y_c - y_p)^2 \leq r$$

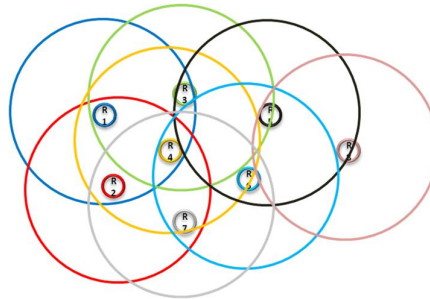


Figura 5.5.: Radio de cobertura de cada nodo [Fuente: Elaboración Propia].

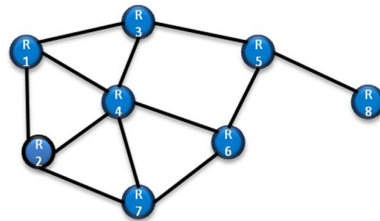


Figura 5.6.: Enlaces virtuales formados por nodos [Fuente: Elaboración Propia].

$$(x_c, y_c) = \text{ubicacion del nodo } c'$$

$$(x_p, y_p) = \text{ubicacion del nodo } p'$$

$$r = \text{radiode cobertura del nodo } c$$

Identificadores de los nodos

Recordemos que cada nodo genera aleatoriamente su propia dirección, la cual consiste en un filtro de Bloom de tamaño m y k hashes (donde $k \ll m$). En la práctica, esto significa que el identificador del nodo es un arreglo de m bits, en el que sólo unos cuantos bits (k bits) tienen el valor de 1.

5.5. Resultados

5.5.1. Consideraciones

Para generar los resultados, se consideran los siguientes factores:

- *Numero de enlaces virtuales'* : Es la cantidad de asociaciones de un nodo con otro, siempre y cuando tengan cobertura entre sí.
- *Rango de cobertura* : Es el área de propagación de un dispositivo inalámbrico para poder establecer una comunicación.
- *Paquetes perdidos por radiación'* : Es la cantidad de paquetes que llegan a otros nodos no participantes en una cierta ruta o grupo.
- *Bucles detectados* : Un bucle ocurre cuando se produce un falso positivo de un paquete, y éste termina volviendo a un nodo anterior en la ruta que viene siguiendo.
- *Ancho de banda desperdiciado* : Es el número de enlaces virtuales que los paquetes recorren en la red, que no siguen su camino correcto.
- *Ancho de banda útil'* : Es el número de enlaces virtuales que los paquetes recorren en la red, sobre sus caminos correctos.

A continuación se muestran los resultados de envíos de paquetes a grupos multicast. Las simulaciones se aplicaron redes de 8 y 14 nodos. En ambos escenarios se hicieron pruebas con distintos tamaños de filtros de Bloom y distintos valores de k aplicados a los filtros. Para la generación de grupos multicast de distribución uniformemente variable, se generaron 50 grupos multicast y se hicieron 5 repeticiones de todas las simulaciones, encontrando el promedio entre los resultados del total de simulaciones.

Usando la red de 8-nodos, considerando que cada nodo tiene un radio de propagación de 22 metros, un resultado importante es que el ancho de banda desperdiciado es menor que el ancho de banda útil usado (ver Figura 5.7). Este ancho de banda desperdiciado son los paquetes duplicados por error (falsos positivos) o por radiación. Los paquetes perdidos por radiación son menores al ancho de banda útil. En consecuencia, la red hace un uso eficiente de los recursos limitados.

Los bucles se producen cuando ocurre un falso positivo de un paquete (ver Figura 5.8) y éste termina volviendo a un nodo anterior en la ruta que viene siguiendo.

En las imágenes que a continuación se muestran, se evaluó el comportamiento de la red variando el rango de cobertura de los nodos. En las pruebas realizadas a la red de 8 – nodos y a la red de 14 – nodos, se varía el radio de cobertura cada 8,5 unidades (pueden ser metros o cualquier otra unidad de medida). La finalidad es poder visualizar mejor el comportamiento de nuestra propuesta. Las redes antes mencionadas se encuentran estructuradas en el Anexo1 como *WArequipa* y *WPeru*, en la clase *cNetwork*.

En la Figura 5.9, se muestra el número de enlaces virtuales (o ficticios) en la red de 8nodos según el radio de cobertura. Al incrementar el radio de cobertura, aumentan la cantidad de enlaces virtuales. Estos enlaces permiten tener conexión entre los nodos. Si el radio de cobertura es menor, el alcance de un nodo se vuelve casi nulo, en cambio, cuando el radio se incrementa, permite que los dispositivos se encuentren y se interconecten con más nodos vecinos.

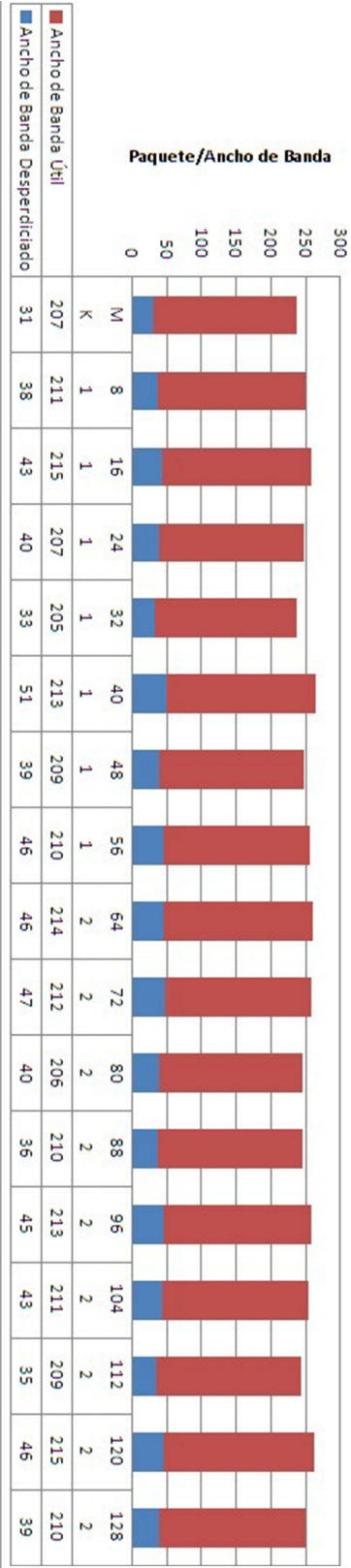


Figura5.7.: Anchodebandaútilydesperdiciado,redde8-nodos[Fuente:ElaboraciónPropia].

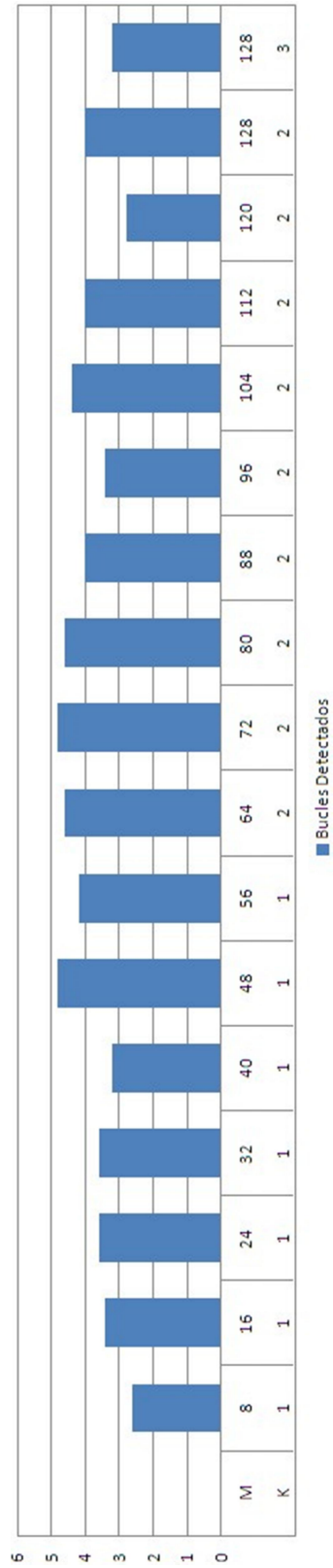


Figura 5.8.: Bucles detectados, redde8-nodos[Fuente:ElaboraciónPropia].

5.5 Resultados

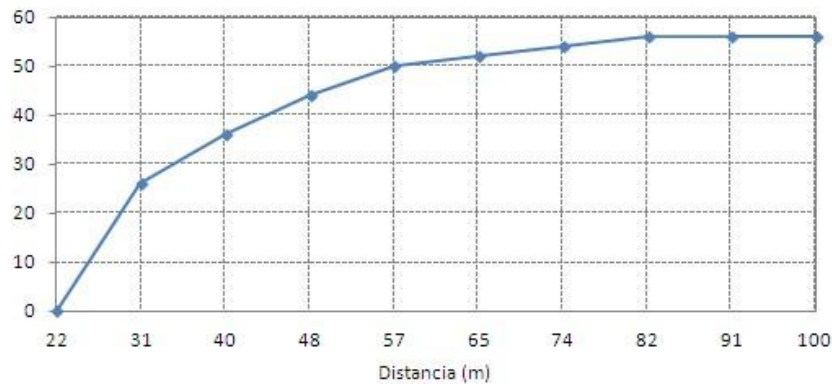


Figura 5.9.: Número de enlaces virtuales, red de 8-nodos [Fuente: Elaboración Propia].

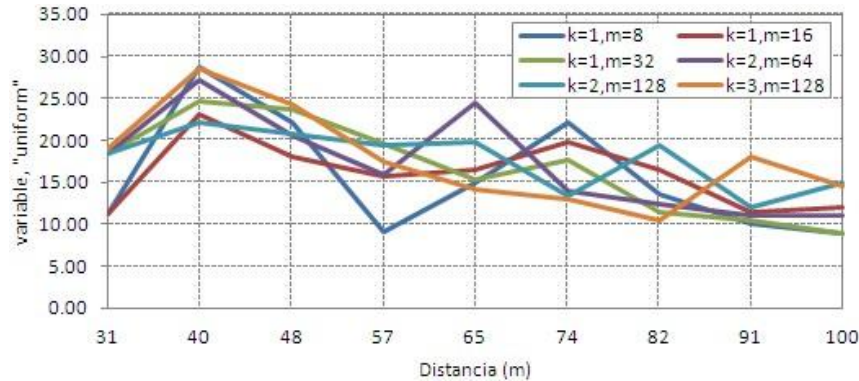


Figura 5.10.: Paquetes perdidos por radiación, red de 8-nodos [Fuente: Elaboración Propia].

En las simulaciones desarrolladas, se muestran los paquetes perdidos por radiación (Ver la Figura 5.10), los cuales varían conforme cambia el radio de cobertura. Además, se ve que al incrementar el área de cobertura, los paquetes perdidos por radiación disminuyen gradualmente. Esto se debe a que los nodos incrementan el número de sus conexiones inalámbricas con sus vecinos. Esto ayuda a generar mejores rutas cuando se generan grupos multicast.

Los bucles en la red de 8-nodos (ver Figura 5.11), se reducen al incrementar el área de cobertura, de acuerdo a los resultados. En la Figura 5.11, se puede observar que cuando se alcanza cierta distancia de cobertura, los bucles desaparecen. Esto se debe a que los nodos ya no requieren realizar muchos saltos entre ellos, para alcanzar a otro, ya que se tiene enlaces largos y directos entre los nodos.

El ancho de banda desperdiciado (Ver Figura 5.12) es el número de enlaces virtuales que los paquetes recorren en la red, y que no siguen su camino correcto. Al tener un número menor de enlaces virtuales aumenta el ancho de banda desperdiciado, cuando

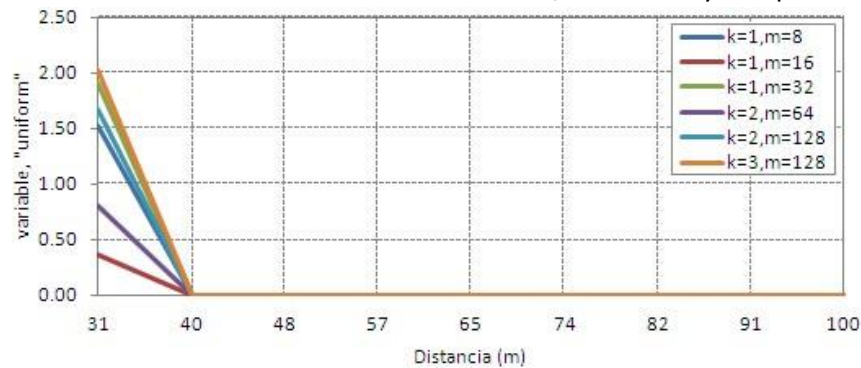


Figura 5.11.: Bucle detectados, red de 8-nodos [Fuente: Elaboración Propia].



Figura 5.12.: Ancho de banda desperdiciado, red de 8-nodos [Fuente: Elaboración Propia].

se incrementa el área de cobertura, se reduce el ancho de banda desperdiciado.

En la Figura 5.13, se observa el ancho de banda útil usado, este es el número de enlaces virtuales que los paquetes recorren en la red, sobre sus caminos correctos. En la Figura 5.13, el uso del ancho de banda se mantiene estable al incrementar el área de cobertura.

En la Figura 5.14 se muestra los resultados de envíos de paquetes a grupos multicast en la red de 14-nodos, con un radio de cobertura por nodo de 22 unidades (metros). Se hicieron pruebas con distintos tamaños de filtros de Bloom y distintos valores de k aplicados a los filtros. Para la generación de grupos multicast de distribución uniformemente variable, se generaron 50 grupos multicast y se hicieron 5 repeticiones de todas las simulaciones, encontrando el promedio entre los resultados del total de simulaciones. Se observa que el ancho de banda desperdiciado es menor que el ancho de banda útil usado (ver Figura 5.14). Este ancho de banda desperdiciado son los paquetes duplicados por error (falsos positivos) o por radiación. Los paquetes perdidos por radiación son menores al ancho de banda útil, en consecuencia, la red hace un uso eficiente de los recursos limitados.

5.5 Resultados

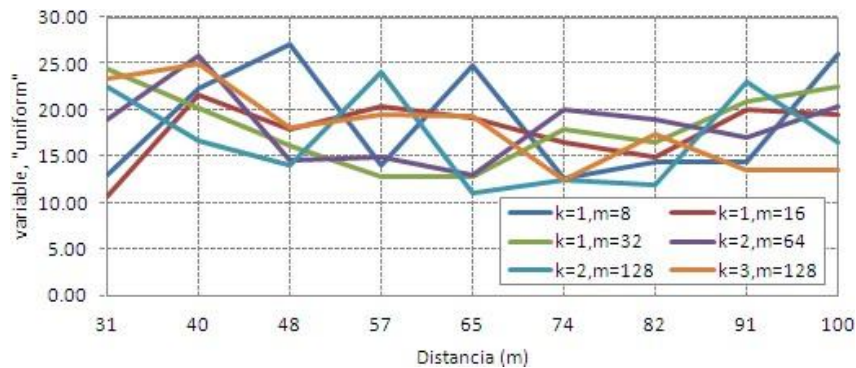


Figura 5.13.: Ancho de banda útil usado, red de 8-nodos [Fuente: Elaboración Propia].

Los bucles se producen cuando ocurre un falso positivo de un paquete (ver Figura 5.15) y éste termina volviendo a un nodo anterior en la ruta que viene siguiendo.

En la Figura 5.16, se muestra el número de enlaces virtuales (o ficticios) en la red de 14-nodos según el radio de cobertura. Al igual que la red de 8-nodos, al incrementar el radio de cobertura, aumenta el número de enlaces. Si el radio de cobertura es menor, el alcance y conexión de un nodo se vuelve casi nulo, en cambio, cuando el radio se incrementa, permite que los nodos puedan conectarse.

En la Figura 5.17, se ve que los paquetes perdidos por radiación son reducidos al incrementar el radio de cobertura. Al comparar los resultados con la Figura 5.10, se ve que el número de paquetes perdidos por radiación es un poco mayor, esto se debe a que hay mayor participación de dispositivos en la red. En las simulaciones se involucra el tamaño del filtro (M), la imagen muestra el comportamiento de cada filtro de manera independiente.

En la Figura 5.18, se observa los bucles detectados en la red de 14-nodos. Estos son reducidos al incrementar el área de cobertura, al tener más enlaces, los paquetes llegan a sus destinos con menor número de saltos. La cantidad de bucles detectados está relacionada directamente a la cantidad de nodos participantes en la red.

En la red de 14-nodos (ver Figura 5.19), se observa que, mientras mayor sea la participación de nodos en la red, mayor será el ancho de banda desperdiciado, pero al incrementar el área de cobertura se reduce el ancho desperdiciado.

En la simulación mostrada en la Figura 5.13 y la Figura 5.20 se muestra que, al incrementar el radio de cobertura, estos mantienen el mismo ancho de banda, además, el incremento del radio logra aumentar el número de enlaces, pero no afecta el ancho de banda útil usado.

En las distintas simulaciones desarrolladas, se verificó que la propuesta de solución es una buena opción para mejorar el protocolo ZigBee, además no solo está orientado en autogenerar las direcciones para cada nodo, también tiene la facultad de generar

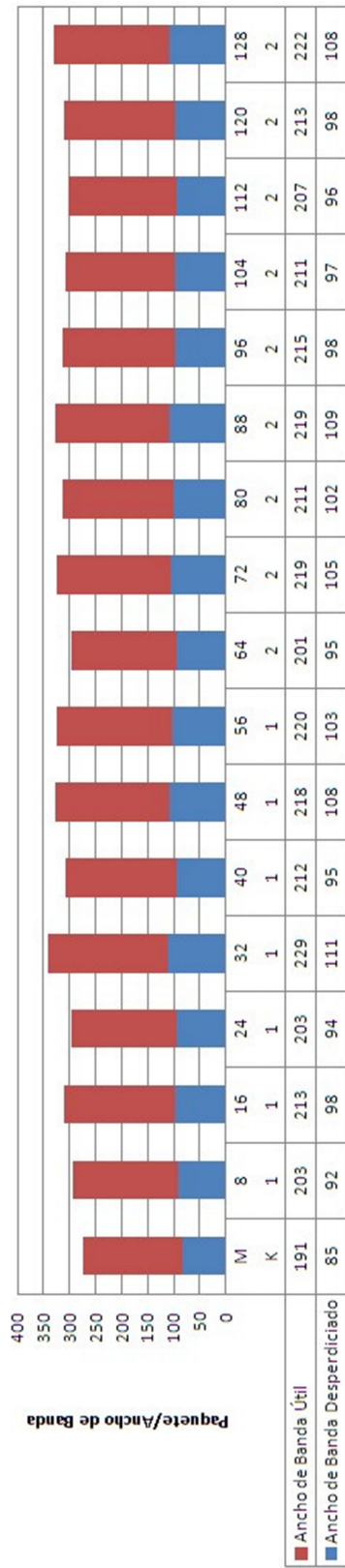


Figura5.14.: Ancho de banda útil y desperdiciado, red de 14 nodos [Fuente: Elaboración Propia].

5.5 Resultados

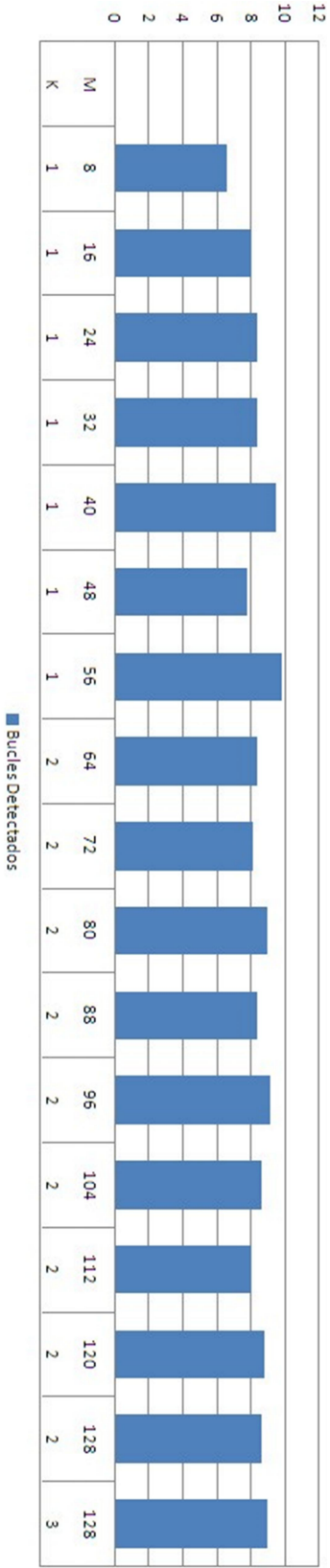


Figura5.15.: Buclesdetectados,redde14-nodos[Fuente:ElaboraciónPropia].

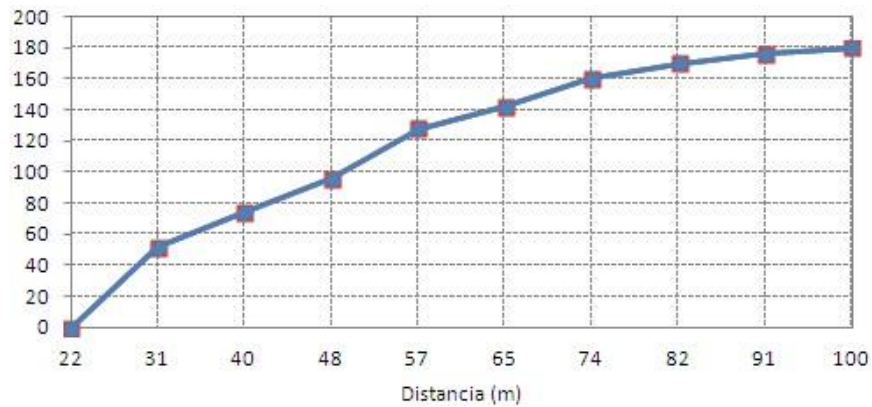


Figura 5.16.: Número de enlaces, red de 14-nodos[Fuente: Elaboración Propia].

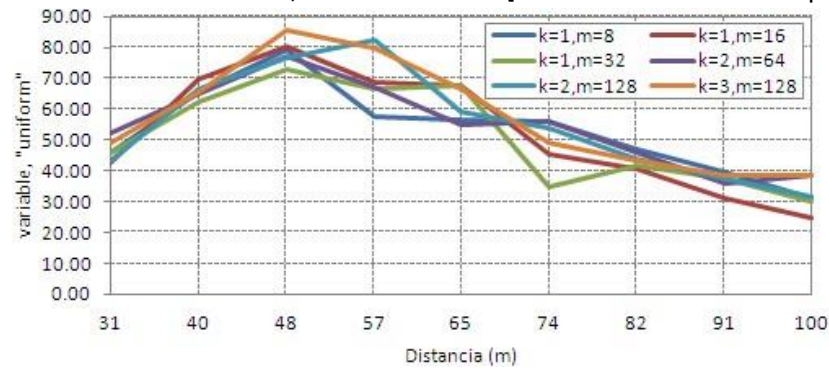


Figura 5.17.: Paquetes perdidos por radiación, red de 14-nodos [Fuente: Elaboración Propia].

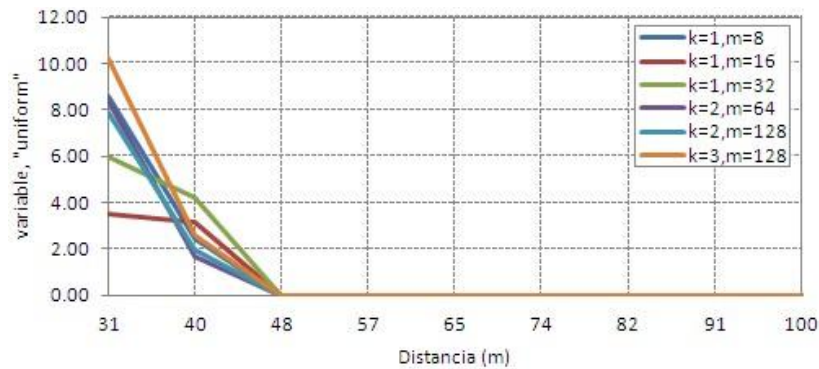


Figura 5.18.: Bucles detectados, red de 14-nodos [Fuente: Elaboración Propia].

resúmenes de nodos hijos y grupos multicast. Si comparamos los resultados con los procedimientos normales del protocolo ZigBee, como se muestra en la siguiente sección, nuestra propuesta contribuye a mejorar algunos aspectos de Zigbee.

5.5 Resultados

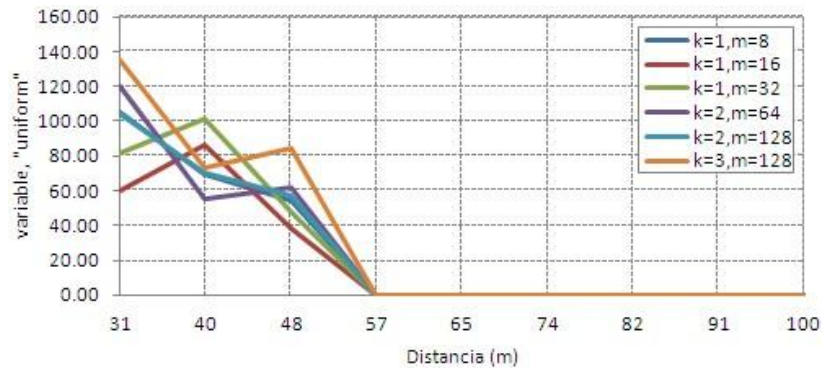


Figura 5.19.: Ancho de banda desperdiciado, red de 14-nodos [Fuente: Elaboración Propia].

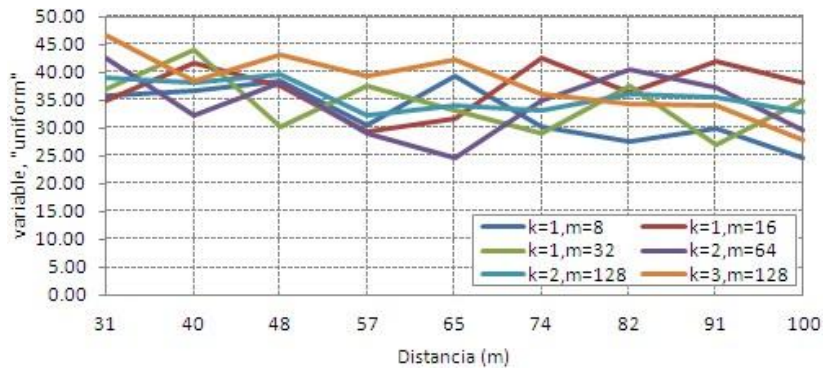


Figura 5.20.: Ancho de banda útil usado, red de 14-nodos [Fuente: Elaboración Propia].

5.6. Comparación con el protocolo ZigBee

En la Tabla 5.3 se muestra un cuadro comparativo, de forma cualitativa, entre el protocolo ZigBee y nuestra propuesta de mejora de Zigbee para las comunicaciones multicast.

Se observa que, al realizar un análisis estructural de nuestra propuesta, se consigue mejorar el funcionamiento de ZigBee para el caso de las comunicaciones con grupos multicast. En las imágenes que a continuación se muestran, se representa un ejemplo que ayuda a ilustrar la comparación entre ZigBee y Zigbee con filtros de Bloom. En la Figura 5.21 se observa el mecanismo típico de ZigBee para el reconocimiento de sus nodos vecinos. Cada nodo presenta una dirección corta de 16 bits, la cual es asignada por un nodo coordinador. En la figura referida se han resumido los pasos que realiza ZigBee para la asignación de direcciones y la determinación de la topología en forma de árbol.

En la Figura 5.22 se representa la asignación de nuevas direcciones de nuestra pro-

ZigBee	ZigBeec/Bloom
Minimum Spanning tree para todas las demandas multicast (a pesar de no es el árbol óptimo para todas las demandas)	Árbol ad-hoc (dedicado) para cada grupo multicast
El coordinador calcula las direcciones y las reparte a hijos y descendientes mayor complejidad	Cada nodo se asigna aleatoriamente una dirección en forma de filtro de Bloom
Los paquetes son enviados de forma centralizada desde el coordinador	El mismo nodo que genera los datos (nodo fuente) envía los paquetes
Se requiere almacenar en todas los grupos y por los nodos destino	Sólo se requiere almacenar los datos por grupo, el resto se encuentra codificado en el filtro de Bloom
Se debe hacer un búscado de cada una de las entradas de la tabla	Se busca sólo el grupo y luego sólo se hace una operación AND

Cuadro 5.3.: Comparación entre el protocolo ZigBee y la propuesta ZigBeec/Bloom [Fuente: Elaboración Propia].

5.6 Comparación con el protocolo ZigBee

Característica
Árbol utilizado para el enrutamiento multicast
Asignación de direcciones
Envío de paquetes
Tamaño de las tablas de enrutamiento
Velocidad de conmutación

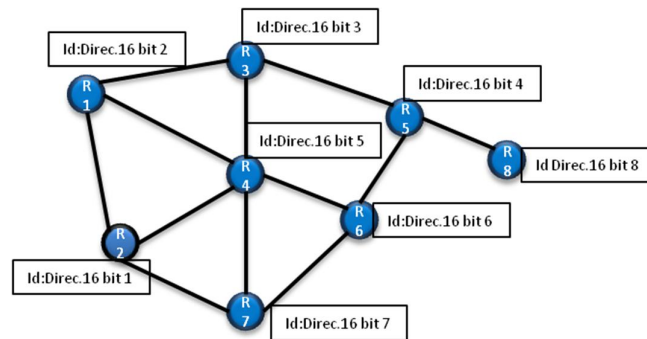


Figura 5.21.: Asignación de direcciones en redes ZigBee [Fuente: Elaboración Propia].

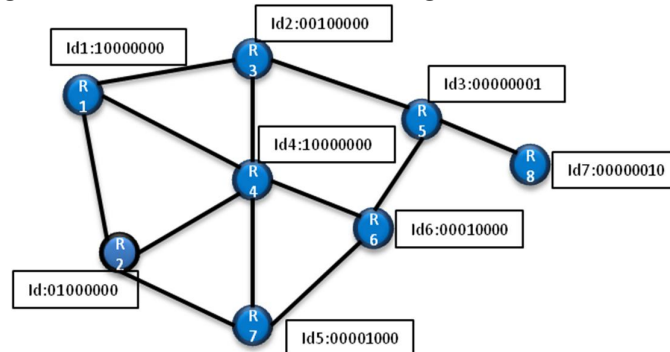


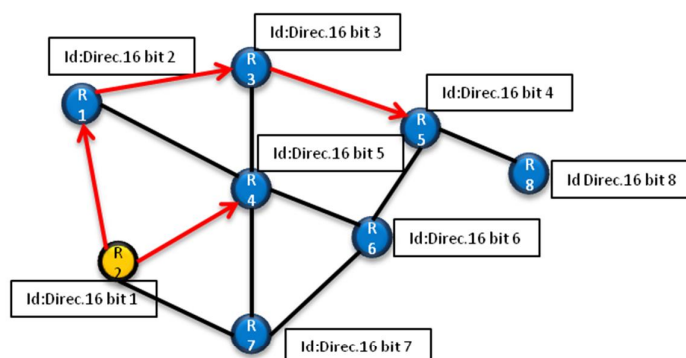
Figura 5.22.: Asignación de direcciones usando filtros de Bloom [Fuente: Elaboración Propia].

puesta, las cuales son almacenadas en forma de filtros de Bloom. Dichas direcciones son autogeneradas por cada nodo y también se representan como arreglos de bits.

La Figura 5.23, representa una red de nodos ZigBee, en la que se observa (resaltada con líneas rojas) una estructura tipo árbol con su nodo coordinador (R2). También se muestran pequeñas tablas de enrutamiento, las cuales albergan nodos vecinos. R2 alberga en su registro todos los nodos participantes, en la figura se puede ver la existencia de entradas para encaminar los paquetes. En ZigBee aún no se ha implementado la difusión multicast, la manera de crear grupos multicast es como se muestra a continuación. Se tiene 4 entradas que involucran los nodos participantes para establecer una difusión multicast, estos son almacenados en las tablas de enrutamiento de cada nodo por grupo multicast formado, esto incrementa el uso de las tablas de enrutamiento por el exceso de entradas.

El proceso de formación de grupos multicast (Figura 5.24) se optimiza al usar filtros de Bloom. En la propuesta se tiene una entrada por todo un grupo multicast. Es más simple almacenar en un solo filtro de Bloom, que albergar varias entradas en una tabla de enrutamiento. En este ejemplo, usando ZigBee se consigna 4 entradas

5.6 Comparación con el protocolo ZigBee



Tablas de enrutamiento

Nodo 2:

G.Multicast

Id:Dir.ec.16 bit 2
Id:Dir.ec.16 bit 3
Id:Dir.ec.16 bit 4
Id:Dir.ec.16 bit 5

G.Broadcast

Id:Dir.ec.16 bit 1
Id:Dir.ec.16 bit 2
Id:Dir.ec.16 bit 3
Id:Dir.ec.16 bit 4
Id:Dir.ec.16 bit 5
Id:Dir.ec.16 bit 6
Id:Dir.ec.16 bit 7
Id:Dir.ec.16 bit 8

Nodo 1:

G.Multicast

Id:Dir.ec.16 bit 2
Id:Dir.ec.16 bit 3

G.Broadcast

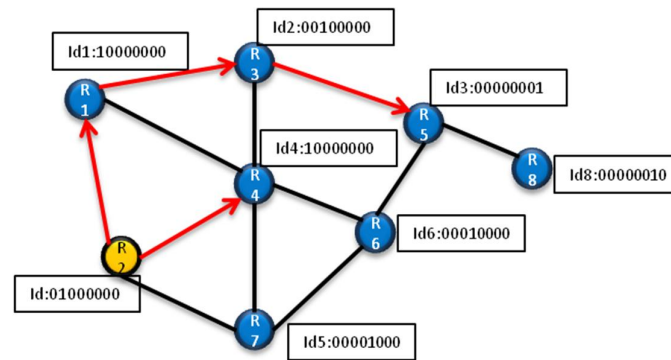
Id:Dir.ec.16 bit 1
Id:Dir.ec.16 bit 2
Id:Dir.ec.16 bit 3
Id:Dir.ec.16 bit 4
Id:Dir.ec.16 bit 5
Id:Dir.ec.16 bit 6
Id:Dir.ec.16 bit 7
Id:Dir.ec.16 bit 8

Figura 5.23.: ZigBee con grupos multicast [Fuente: Elaboración Propia].

cuando se forma un grupo multicast, mientras que en la propuesta de filtros de Bloom sólo fue necesario el uso de una entrada en un filtro, luego se realizó la operación *OR*, para almacenar todos los nodos en una sola entrada. Además, cuando se requiera establecer una difusión de tipo broadcast no es necesario tener varias entradas para realizar dicha petición.

En la Figura 5.25 se representa el caso en que ZigBee requiere enviar un paquete a un grupo multicast específico, cuando se tiene consignado el grupo multicast al cual enviar el paquete. El paquete pasa por distintos nodos realizando la verificación, y si realmente es el nodo al cual debe difundir el mensaje (como se ve en la Figura 5.25), el paquete revisa cada entrada almacenada en el grupo multicast que se desea difundir el mensaje. Si dicha verificación resulta positiva, se distribuye el mensaje. Este proceso realiza la verificación por cada entrada que tenga almacenado el nodo multicast.

En el procedimiento mostrado en la Figura 5.26 se demuestra que, haciendo uso de los filtros de Bloom, se reduce el número de verificaciones hechas en la tabla de enrutamiento. La verificación consiste en una operación *AND* de la cabecera del paquete con el filtro resumen almacenado en el nodo, previa identificación del grupo multicast en el que se debe difundir el paquete (cada grupo multicast tiene una topología diferente, por lo que cada nodo tiene un filtro que resume todos sus hijos en ese grupo/árbol multicast). Si el resultado es positivo, el paquete se difunde a través de la interfaz de radio, de lo contrario se descarta. En comparación con



Tablas de enrutamiento

Nodo 2:

G.Multicast

Id: 10100001

G.Broadcast

Id: 10111011

Nodo 1:

G.Multicast

Id: 00100001

G.Broadcast

Id: 10111011

Nodo 3:

G.Multicast

Id: 00000001

G.Broadcast

Id: 10111011

Nodo 5:

G.Multicast

Id: 00000000

G.Broadcast

Id: 10111011

Figura 5.24.: Uso de filtros de Bloom en la formación de grupos multicast [Fuente: Elaboración Propia].

ZigBee, el tiempo de verificación se resume a unas pocas entradas, y no a varias, como lo realizaría ZigBee.

A continuación se realiza el análisis, respecto al número de entradas en la tabla de enrutamiento que tendría ZigBee acorde a la cantidad de nodos involucrados en una red. Esto es comparado con nuestra propuesta de solución para corroborar la eficiencia de los filtros de Bloom.

Se han considerado las siguientes variables:

G_m : Número promedio de grupos multicast en los que un nodo participa

G : Número de grupos multicast N :

Número de nodos de la red

n_m : Número de nodos promedio por grupo multicast $numEntries_{ZigBee}$: Número de entradas en la tabla de enrutamiento en ZigBee $numEntries_{Bloom}$: Número de entradas en la tabla de enrutamiento en la propuesta con filtros de Bloom

■ En ZigBee, para hallar el número total de entradas, se usa la siguiente fórmula:

5.6 Comparación con el protocolo ZigBee

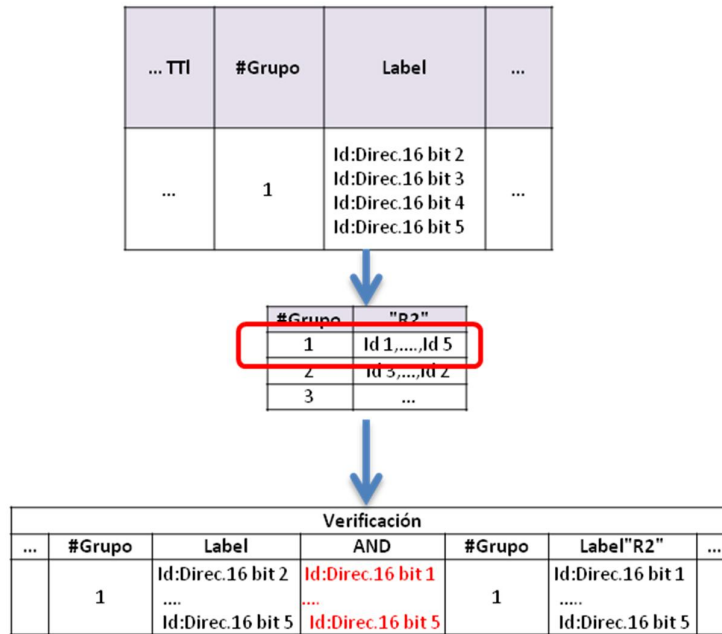


Figura 5.25.: Envío de un paquete a un grupo multicast en redes ZigBee [Fuente: Elaboración Propia].

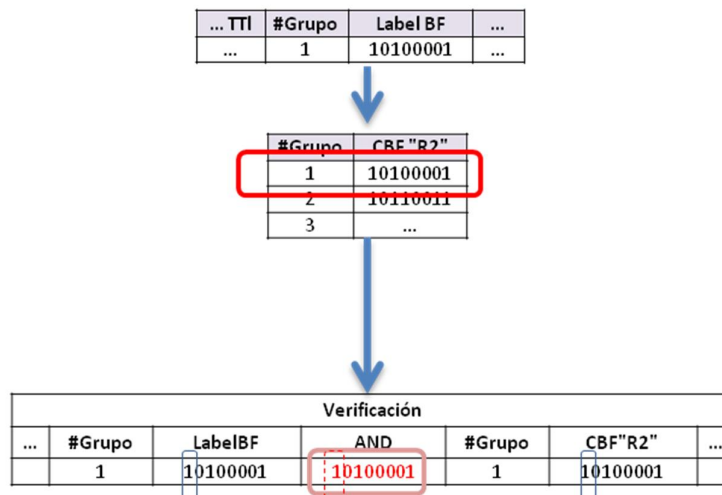


Figura 5.26.: Envío de un paquete a un grupo multicast con filtros de Bloom [Fuente: Elaboración Propia].

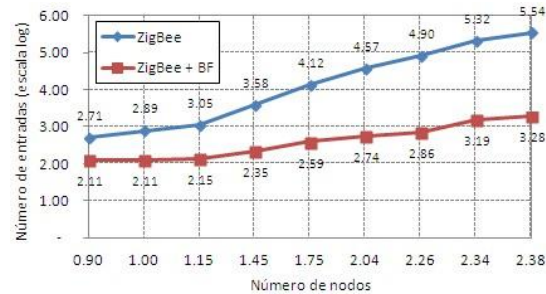
$$numEntries_{ZigBee} = N \cdot G_m(n_m - 1)$$

■ En nuestra propuesta con filtros de Bloom, el número total de entradas es hallada por:

Cuadro 5.4.: Valores simulados en relación a los grupos multicast fijos [Fuente: Elaboración

G	20	20	20	20	20	20	20	20	20
N	8	10	14	28	56	110	180	220	240
n_m	5	7	9	18	35	69	112	138	180
$numEntries_{ZigBee}$	512	780	1120	3808	13328	37400	79920	210980	343680
$numEntries_{Bloom}$	128	130	140	224	392	550	720	1540	1920
G_m	16	13	10	8	7	5	4	7	8

Propia].

**Figura 5.27.:** Total de número de entradas en toda la red [Fuente: Elaboración Propia].

$$numEntries_{Bloom} = N \cdot G_m(1)$$

La Tabla 5.4 muestra los valores correspondientes obtenidos dependiendo de las variables, considerando la cantidad total de número de entradas en ZigBee y filtros de Bloom.

En la Figura 5.27 se observa la efectividad al usar filtros de Bloom, en comparación a ZigBee. En ZigBee se tiene una cantidad de entradas por grupo multicast, en la propuesta con filtros de Bloom, el número de entradas que albergara la tabla de enrutamiento no es mayor a uno por grupo multicast. De esta manera se reduciría en gran medida el número de entradas por nodo, y esto lo convierte en una buena propuesta para aumentar su eficiencia y su escalabilidad en este tipo de redes.

5.7. Conclusiones

- En el esquema desarrollado para mejorar el protocolo ZigBee, se recomienda realizar pruebas con estructuras estáticas y redes pequeñas. Se asignó manualmente las direcciones a cada nodo, luego se realizó la simulación, los resultados obtenidos lograron corroborar los valores obtenidos manualmente. Cuando se trabaja con

5.6 Comparación con el protocolo ZigBee

estructuras estáticas, es más simple administrar redes pequeñas o medianas, y también se pueden realizar pruebas manuales que corroboren el

5.7 Conclusiones

adecuado funcionamiento. En estructuras aleatorias y de mayor cantidad de nodos, es necesario automatizar toda la simulación.

- En el trabajo presentado, se concluye que el uso de filtros de Bloom ha introducido una mejora al protocolo ZigBee para las comunicaciones de grupos multicast. Esto se debe a que cada nodo genere su propia dirección y a que las direcciones son la base de los nuevos filtros resúmenes, los cuales se integran en las tablas de enrutamiento.
- En el protocolo ZigBee las comunicaciones giran alrededor de un mismo nodo coordinador, en cambio en nuestra propuesta es posible establecer nodos orígenes de las peticiones multicast distintos para cada escenario.
- Con nuestra propuesta es posible atender peticiones multicast. El protocolo ZigBee no tiene habilitada esta función, y para realizar el multicast, solo se tiene la alternativa de utilizar el protocolo IP multicast.
- En las pruebas a las que sometimos nuestra propuesta, se demostró la reducción de números de entradas en las tablas de enrutamiento de los nodos. Adicionalmente, cuando se difunde un paquete multicast, el tiempo de operación en ZigBee se ve incrementado por la cantidad de nodos involucrados y por la cantidad de entradas en las tablas de enrutamiento de los nodos.
- Se observa que, al aumentar el radio de cobertura, se logran establecer enlaces con varios dispositivos de un solo salto en la red, reduciendo el ancho de banda desperdiciado, los bucles detectados, y los paquetes perdidos por radiación.
- El número total de entradas que tendría ZigBee en cada nodo es superior a las entradas de nuestra propuesta, y las cantidades de entradas almacenadas en las tablas de enrutamiento crecen al aumentar los dispositivos en la misma red.

Podemos afirmar que el protocolo ZigBee puede ser mejorado con el uso de filtros de Bloom, extendiendo sus capacidades hacia el servicio multicast. Este nuevo esquema de enrutamiento permitiría que la tecnología ZigBee aproveche mejor la red.

6. Conclusiones y Trabajos Futuros

6.1. Conclusiones

En este trabajo hemos propuesto una mejora al protocolo ZigBee, para que pueda soportar un nuevo esquema de enrutamiento mediante filtros de Bloom, con el fin de soportar peticiones multicast y aligerar las tablas de enrutamiento de los nodos. En este sistema, los nodos autogeneran sus propias direcciones de manera aleatoria. Utiliza resúmenes de filtros de Bloom para el reenvío de paquetes a través de los árboles multicast se ha mejorado el rendimiento de la red. Una de las principales ventajas de esta propuesta es que la formación de grupos multicast permite establecer nuevas redes árbol, en las cuales no siempre se mantendrá un mismo nodo como el coordinador para establecer algún tipo de comunicación o difusión de algún paquete, sino que cualquier nodo puede fungir de coordinador, en caso que sea el nodo fuente de la petición multicast.

En las pruebas desarrolladas se hicieron evaluaciones respecto a la cantidad de enlaces virtuales que se generan en la red, al radio de cobertura, al ancho de banda desperdiciado, a la cantidad de bucles detectados, a los paquetes perdidos por radiación y al ancho de banda útil. En comparación con ZigBee, se ha comprobado que la cantidad de entradas en la tabla de enrutamiento que se genera en los nodos es mucho menor. Esto coincide también con los resultados de la fórmula propuesta que determina la cantidad de entradas en la tabla de enrutamiento, con la que se concluye que en redes Zigbee, al incrementar el número de nodos, la tabla de enrutamiento se incrementará proporcionalmente. Sin embargo, cuando trabajamos en una red ZigBee con filtros de Bloom, el número de entradas ingresadas en la tabla de enrutamiento será menor, debido a que éstas se apilan en un arreglo de bits. Este mejoramiento permite reducir el número de entradas en la tabla de enrutamiento portada en cada nodo.

Otra de las grandes ventajas de la adaptación de Zigbee con filtros de Bloom es que cada transmisión multicast puede ser asumida desde el mismo nodo que genera el flujo multicast (nodo origen), sin necesidad de pasar toda la comunicación a través del nodo coordinador. Como se recuerda, en ZigBee no es posible la transmisión individual sin antes pasar por el nodo coordinador central.

6.2. Trabajos Futuros

Para trabajos futuros, sería importante estudiar el problema de tener rutas redundantes mediante el uso de filtros de Bloom, para efectos de mayor resiliencia de las redes. Esto ayudaría a tener rutas alternas en caso de romperse alguna comunicación con algún nodo

que se desea alcanzar. Además, así se evitaría que la red vuelva a reiniciarse para construir una nueva red con los dispositivos que encuentra en su radio de cobertura.

Otro trabajo futuro sería el llegar a simular una red más extensa con nodos dinámicos (desconexión de nodos, nodos móviles, etc.), con el fin de ver cómo una red interactúa con estos nodos, y si se permite alguna forma de mejorar la red.

También, sería importante estudiar qué inconveniente ocasiona tener nodos clones, es decir, nodos con iguales características físicas e iguales funciones. Una pregunta en este tipo de red es: ¿será posible administrar una red inmensa con dos dispositivos idénticos para reducir el trabajo funcional en toda la red?.

Otro problema a estudiar es el de los problemas de seguridad que se puedan derivar del uso de filtros de Bloom en redes Zigbee. Por ejemplo, un nodo malintencionado podría insertar en las cabeceras de sus paquetes filtros de Bloom en forma de arreglos de bits llenos de 1's, lo cual ocasionaría que la red genere un flujo grande de falsos positivos, trayendo como consecuencia el incremento del ancho de banda desperdiciado.

A. Anexo

A.1. Anexo de Código fuente

En la siguiente documentación se anexa código fuente de las simulaciones realizadas.

Bibliografía

- [AN10] Ivan Stojmenovic Amiya Nayak. *Wireless Sensor And Actuator Networks: Algorithms And Protocols For Scalable Coordination And Data Communication*. A John Wiley & Sons, INC., 2010.
- [AR12] Silvia Santini Sebastian Zoller Ralf Steinmetz Andreas Reinhardt, Olivia Morar. Cbfr: Bloom filter routing with gradual forgetting for treestuctured wireless sensor networks with mobile nodes. In *Wireless Sensor Networks Lab, Tecnische Universitat Darmstadt, Germany*, 2012.
- [AZN10] M Sarela S Ruponen J Kempf A Zahemszky, P Jokela and P Nikander. Mpss: Multiprotocol stateless switching. In *in Proceedings of the IEEE Conference on Computer Communications INFOCOM Workshops*, pages 1–6. San Diego, 2010.
- [BG08] Schaefer G Vetterli M Couach O Parlange M. Barrenetxea G, Ingelrest F. Sensorscope: out-of-the-box environmental monitoring. In *The ACM/IEEE 7th International Conference on Information Processing in Sensor Networks (IPSN 2008)*, 2008.
- [BK07] Tapio Suihko Bilge Kartal, Radosveta Sokullu. A new multicast routing algorithm for zigbee networks. pages 146–150, 2007.
- [Blo70a] B.H. Bloom. *Space/Time Trade-offs in Hash Coding with Allowable Errors*, volume 13. 1970.
- [Blo70b] Burthor H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7): 422–426, 1970.
- [C07] Valverde Rebaza J C. *Estandar Inalambrico ZigBee*. PhD thesis, Universidad Nacional de Trujillo, 2007.
- [CAO08] Leslie Eduardo U.S. Carlos Alberto O.H, Deyanira del S.R. Trabajo de tecnicas de alta frecuencia taf-zigbee. Technical report, Universidad Nacional De IngenierÃa (UNI), 2008.
- [Dar10] C. Dargie, W. y Poellabauer. *Fundamentos de redes inalambricas de sensores: teoria y practica*. 2010.
- [dJGD10] Antonio de Jesus Garcia Dominguez. *Recoleccion de Datos Usando Filtros de Bloom en Redes de SeSensor Inalambricos*. Universidad Nacional Autonoma de Mexico, 2010.

Bibliografía

- [DS07] Kamath A Nayak A Stojmenovic I. Das S, Liu H. Localized movement control for fault tolerance of mobile robot networks. In *Proceedings of the First IFIP International Conference on Wireless Sensor and Actor Networks(WSAN 2007)*, 2007.
- [DS09] Nayak A Stojmenovic I. Das S, Liu H. A localized algorithm for biconnectivity of connected mobile robots. *Telecomm Syst*, 40: 129–140, 2009.
- [DT03] Langendoen K. Dam TV. An adaptive energy efficient mac protocol for wsns. In *Proceedings of ACM Sensys*, 2003.
- [EC01] 8000 W. Sunrise Blvd. M/S 2141 Plantation FL 33322 Ed Callaway, Motorola. *Cluster Tree Network*. IEEE P802.15, wireless personal area networks edition, 30 April 2001.
- [Ecl13] Eclipse, Locationtech,mam,Polarsys,eclipse LTS, Eclipse. <http://www.eclipse.org/>.
- [Gar09] Gonzalo Campos Garrido. *Caracterizacion del consumo en redes ZigBee/802.15.4*, chapter 2, pages 11–33. Escuela Tecnica Superior de Ingenieria de Telecomunicaciones, 2009.
- [GM12] David Larrabeiti Gonzalo M.Fernández. Depth-wise multi-protocol stateless switching of multicast traffic. *IEEE*, pages 1–13, 2012.
- [Her13] Jairo Javier Sanchez Hernandez. *Enrutamiento probabilistico para redes tolerantes al retardo*. Instituto Politecnico Nacional, 2013.
- [HL10] and Ivan Stojmenovic Hai Liu, Amiya Nayak. *Applications,Models,Problems,and Solution Strategies*, chapter 1, pages 1–30. Wiley, 2010.
- [IF06] Stojmenovic I. Ingelrest F., Simplot-Ryl D. *Routing and broadcasting in hybrid ad hoc and sensor networks*. Taylor & Francis Group, 2006.
- [JL03] Hill JL. *System Architecture for Wireless Sensor Networks*. PhD thesis, UC Berkeley, 2003.
- [JL10] Ning Tang Jisheng Sui Jianpo Li, Xuning Zhu. Study on zigbee network architecture and routing algorithm. In *2nd international Conference on Signal Processing Systems (ICSPS)*, 2010.
- [JS] hong Wang Xiaofen Zhang Jing Sun, Zhongxia Wang. Research on routing protocols based on zigbee network. Technical report, Shenyang Normal University, Shenyang China, 2 JiLin Normal University, Siping China.
- [LC09] C.R.Jimenez Yedra L.F.falconi Cepeda. *Estudio e implementacion de domotica activado por comandos de voz y comunicacion en ZigBee*,. PhD thesis, Escuela superior Polit cnica de Chimborazo, 2009.

Bibliografía

- [Mic13a] Microchip. *Microchip AN965 -Microchip Stack For the ZigBee Protocol*. Microchip, an965 edition, 04 2013. Microchip Stack For the ZigBee Protocol.
- [Mic13b] Java, ORACLE, Sun Microsystems. <http://www.java.com/>.
- [OG10] Omar Cheikhrouhou Mhamed Abid Olfa Gaddour, Anis Kouba. A multicast routing mechanism in zigbee cluster-tree wireless sensor networks. *College of Computer and Information Sciences, Distributed Computing Systems Workshops*, pages 1–9, 2010.
- [PJ04] Culler D. Polastre J, Hill J. Versatile low power media access for wireless sensor networks. In *Proceedings of ACM Sensys*, 2004.
- [RP06] Zou You-min Ran Peng, Sun Mao-heng. Zigbee routing selection strategy based on data services and energy-balanced zigbee routing. In *IEEE AsiaPacific Conference on Services Computing 6*, 2006.
- [RPL05] Pedro B Velloso Marco D D Bicudo Rafael P Laufer, Igor M Moraes. Negacion de servicio: Ataques y contramedidas. Minicursos do Simposio Brasileiro de Seguranca da Informacao e de Sistemas Computacionais, Florianopolis, Brazil, Setiembre 2005.
- [Sar12] Prativa P Saraswala. A survey on routing protocols in zigbee network. *International Journal of Engineering Science and Innovative Technology (IJESIT)*, 2: 1–8, 2012.
- [SI5a] Simplot-Ryl D Nayak A Kuruvila J. Stojmenovic I, Ingelrest F. Design guideline for routing protocols in ad hoc and sensor networks with a realistic physical layer. *IEEE Commun Mag*, 43: 101–106, 2005a.
- [Soc13] 802.15.4-2003 - ieee standard for telecommunications and information exchange between systems - lan/man specific requirements part 15: Wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (wpan), IEEE Standars Association, IEEE Computer Society, IEEE. <http://standards.ieee.org/findstds/standard/802.15.4-2003.html>.
- [Soh07] Minoli D. Znati-T. Sohraby, k. *Redes de sensores inalÃ¡mbricas: La tecnologÃ¡a, los prprotocol y aplicaciones*. 2007.
- [TK14] Jinyoung Yang Seong-eun Yoo Taehong Kim, Seong Hoon Kim. Neighbor table based shortcut tree routing in zigbee wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 25: 706–716, 2014.

- [TO08] Norihiro Ishikawa Tomoyuki Osano, Yoshitaka Uchida. Routing protocol using bloom filters for mobile ad hoc networks. *IEEE Computer Society*, 2008.
- [YW04] Estrin D. Ye W, Heidemann J. Medium access control with coordinated adaptive sleeping for wsns. *IEEE/ACM Trans Network*, 12: 493–506, 2004.

Bibliografia

- [Zig13] ZigBee. Zigbee alliance, control your world, 2013.

Nomenclatura

AODV	Ad hoc On Demand Distance Vector
BS	Estación Base
CBFR	Counting Bloom Filter Routing
CBFs	Couting Bloom Filters
CDS	Conjunto Dominante Conectado
CH	Cluster Head
CRC	Cyclic Redundancy Check
CTP	Collection Tree Protocol
D-MPSS	Depth-Wise Multi-Protocol Stateless Switching
DD	Designated Device
FFD	Full Function Device
MAC	Control de Acceso al Medio
MPLS	Multiprotocol Label Switching
MST	Minimum Spanning Tree
PHY	Physical Layer
RFD	Reduced Function Device
RREP	Route Reply
RREQ	Route Request
STR	Shortcut Tree Routing
UAV	Vehículo Aéreo No Tripulado
UDG	Grafo de Disco Unitario

Nomenclature

WSANs	Wireless Sensor Actuator Networks
WSN	Redes de Sensores Inalámbrico

ZC	ZigBee Coordinator
ZED	ZigBee End Device
ZR	Router ZigBee
ZTR	ZigBee Tree Routing